Development and evaluation of a multiscale keypoint detector based on complex wavelets

Pashmina Bendale
Churchill College, Cambridge

January 2011

Typeset using the LaTeX document preparation system.

Signal Processing and Communications Laboratory,
Department of Engineering,
University of Cambridge,
Trumpington Street,
Cambridge, CB2 1PZ, U.K.

Declaration

The research described in this dissertation was carried out by the author between October 2006 and October 2010. Where reference has been made to the work of others, this is acknowledged in the text and bibliography. This dissertation has not been submitted in whole or part for a degree at any other university. Its length, including footnotes, appendices and bibliography is less than 65,000 words and it contains less than 150 figures.

Pashmina Bendale

Keywords

Keypoint detector, keypoint descriptor, complex wavelets, scale-space, multiscale epipolar geometry

Summary

This thesis develops a multiscale keypoint detector and descriptor based on the Dual-Tree Complex Wavelet Transform (DTCWT). First, we develop a scale-space framework called the 4S-DTCWT that uses the dyadic decomposition of the DTCWT but achieves denser sampling in scale by interleaving several DTCWT trees, leading to reduced scale-related aliasing. This forms the foundation for the rest of our work. Then, we present a new DTCWT based keypoint detector (BTK), which exhibits improved spatial localisation owing to the use of a more selective cornerness measure and keypoint localisation in individual levels in the 4S-DTCWT. A number of scale refinement approaches are investigated.

The improved keypoint position and scale localisation directly leads to more robust image characterisation using DTCWT based visual descriptors. We also present some ways of speeding up both the descriptor and the matching computations. These changes make it possible to use the system in practical scenarios.

We develop a novel, fully automated framework for the evaluation of keypoint detectors and descriptors. This includes a new dataset containing 3978 calibrated images from 2 cameras of 39 different toy cars on a turntable. The dataset, calibration images, inter-camera calibration, rotational calibration and test scripts are publicly available. We establish ground truth correspondences using a three-image setup, with fixed angular separation between two of the three views, thus reducing the dependency on angular separation when compared to conventional epipolar line search.

Various keypoint detectors and descriptors were compared with DTCWT based methods using this framework. To the extent possible, we separated

the evaluation of the keypoint detectors from that of the descriptors. The main conclusions were that DTCWT based methods can achieve a performance comparable, if not superior, to that of established methods. We also showed that, although repeatability of keypoint detections falls off reasonably steeply with change in viewing angle, conditioned on an associated keypoint being detected at a reasonably correct corresponding location, descriptor similarity is hardly affected by viewpoint variation.

Finally, we show how an evaluation that is based purely on the prior knowledge of the geometry of the scene can be useful in eliminating the inaccuracies involved in appearance based evaluations. This uses an enhanced epipolar constraint that exploits both positions and scales of keypoints to constrain the range of possible matches.

Acknowledgements

I am grateful to my supervisor, Prof. Nick Kingsbury for providing me with an opportunity to study in Cambridge. Nick has always encouraged me to learn on my own and given me ample freedom to try new things. He has taken keen interest in my progress and supported my initiatives and provided helpful comments when needed. I appreciate his questioning me at every stage, his constructive criticism and timely warnings. I would particularly like to thank him for providing me with prompt feedback at very short notice.

I am immensely thankful to Dr. Bill Triggs. Bill has provided excellent advice, rigourous training and continual support. He has been a motivational force for research as well as life. Working with him has been a great learning experience and I consider myself fortunate to have had a chance to work with him. He has influenced not only the content of this thesis but also its presentation. I am also thankful to him for funding my visits to Grenoble. His eternal optimism and great sense of humour has made it easier to get past difficult times (and get more done!). He taught me to deal with things as they came and to enjoy the process of scientific research without getting bogged down due to temporary failures.

A special thank you to my examiners Prof. Roberto Cipolla and Prof. David Bull for agreeing to conduct my viva at short notice. Thanks also to Prof. Bull for travelling all the way in heavy snow and to Prof. Cipolla for encouraging me to look at the big picture!

I am especially thankful to Dr. Jonathan Cameron, my friend and collaborator. He has been very helpful, sometimes unknowingly, at the most crucial times. Company on evenings and weekends during the writing of this document is gratefully acknowledged. Thanks also for boosting my spirits

with sparkling conversations (and hot chocolates). I am most thankful to him for helping me stay sane and happy. Thanks also for carefully proof-reading the dissertation multiple times. Thanks to Dr. Simon Hill for company, conversations and advice over the last few years. Thanks to friends Larry and Tee for company on many trips and for fun-filled conversations.

I am deeply indebted to the Gates Cambridge Trust for their generous financial support.

Finally, I wish to thank my parents and my brother for constantly encouraging me, making me believe in myself, for giving me the strength to move on and for all the silent sacrifices they have made for me through the years.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

For humans, recognising objects in visual scenes is an important everyday task that we perform effortlessly. When we see an object for the first time, we automatically identify and memorise its unique identifying visual properties. When we see it again, we try to match its observed properties with those of familiar or remembered objects. A similar approach has proved successful for the automation of the process of image based object recognition on a computer. In order to recognise a previously seen object, the computer has to find its characteristic patterns in the image, and then match these against a database of stored object patterns. Such characteristic patterns are commonly known as *visual features*. Features may be global or local. Global ones are most useful for analysing scenes as a whole, while local ones are useful for analysing parts of a scene giving more robustness to variations such as occlusions, changes in viewpoint *etc.* The rapid development of digital imaging has created many applications for object recognition and hence a need for the development of fast and reliable approaches for the automatic extraction and matching of local features. Local features provide a compact representation of characteristic patterns, which allows the efficient analysis of large numbers of images. Besides being compact, such representations also need to be distinctive enough to aid rapid recognition and discriminative enough to avoid confusion between similar objects.

In this thesis, we focus on the problem of the detection and description of

local features in images. We begin by describing the problem in more detail and outlining the challenges involved in solving it. We then summarise our work and provide an overview of subsequent chapters.

1.1 Problem definition

One of the fundamental problems in object recognition and retrieval is that of matching image elements. Typically, we might want to search for an object or a part of an object in thousands of images. A powerful approach to this is to detect *keypoints* and match their image neighbourhoods [Lowe, 1999, Schmid and Mohr, 1997]. Keypoints are distinctive local image regions that can be consistently detected despite a range of transformations of the image. *Keypoint descriptors* encode the keypoint's image neighbourhood in a manner that is also invariant under these transformations [Lowe, 2004, Mikolajczyk and Schmid, 2005]. Keypoints and their descriptors, together called local features, tell us what is distinctive in the image and where it is located. Not only does this greatly reduce the amount of data that the system has to handle, it also retains the most informative parts of the image, thus facilitating fast search and index in large databases. Local features also provide suitable inputs for simple learning techniques for automated recognition, classification and identification applications.

As local features must be recovered and matched consistently in the presence of image variation such as noise, changes in illumination and viewpoint, the main challenges involved in defining and using them are:

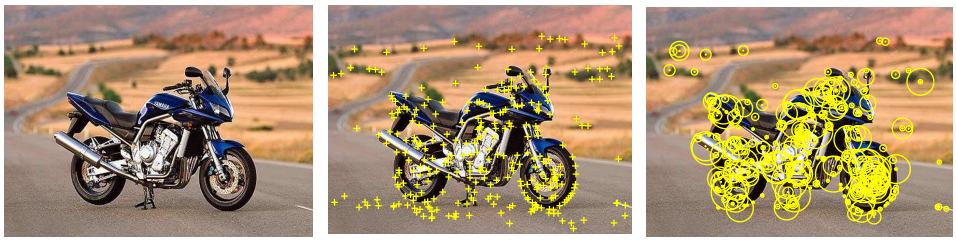


Figure 1.1: An example image, the detected keypoint locations and scales shown on the image. This image is a part of the CalTech Bike dataset [CalTech, 2001].

Repeatable localisation in space. Keypoints are defined by a spatial location, a scale and possibly a local orientation in the image (*c.f.* Figure 1.1). They are usually located at or in the vicinity of a point at which some image property changes significantly in two or more directions. The image may be affected by one or more transformations such as imaging noise, illumination changes and viewpoint variation. The latter incorporates scale changes, translation, in-plane rotation and out-of-plane rotation, all of which can be approximated locally as affine transformations (at least for locally near-planar surfaces). In the presence of such transformations, a keypoint should ideally be detected in the exact corresponding location in the transformed image. Any shifts in the location of the detected keypoint may cause the image neighbourhoods (descriptors) to fail to match correctly.

Repeatable localisation in scale. In order to match objects that appear at any scale in the image, multiscale keypoints are required. In these, not only the position of the keypoint, but also its image scale, must be distinctive. Each keypoint thus associates a scale with its position. The scale measures the extent of the image neighbourhood of the keypoint that must be encoded in order for the keypoint to be matched correctly. Corresponding keypoints must correspond at any relevant transformation in both position and scale. The scale estimates must be repeatable enough to give reliable matching and descriptions.

Robust local description. Given an accurate and stable keypoint in an image, we need to encode its image neighbourhood in a robust way. The description must be discriminative enough to be able to distinguish between visually different feature neighbourhoods, but consistent enough to provide similar descriptions for similar neighbourhoods across different views of the same object or different object instances. It must also be reasonably robust to position and scale localisation errors in the detector.

These are generic requirements for local features. However, the details vary depending on the application. For example, in the case of keypoints

used in camera calibration with chess board patterns [Strobl et al., 2005], localisation accuracy is of prime importance while scale estimation is not very important because the input consists only of corners and typical calibration toolboxes do not use the scale of the keypoints at all. In object recognition applications [Schmid and Mohr, 1997, Lowe, 2001, Mikolajczyk and Schmid, 2001], the distinctiveness of the keypoint is as important as the accuracy of its spatial localisation. In mobile phone applications of object recognition [Lepetit and Fua, 2006, Rosten et al., 2010], the simplicity and speed of keypoint matching is paramount, whereas in offline panorama stitching [Brown and Lowe, 2007, Schaffalitzky and Zisserman, 2002], the accuracy of keypoint matching is most important, whereas speed is of secondary importance because the process is run only once. Our work focuses on the generic requirements. We shall use the term *feature characterisation* to include keypoint detection, keypoint description and keypoint descriptor matching in big databases.

1.2 Contributions

This project aims to solve two interlinked problems: The first is that of accurate, repeatable feature characterisation. There exist other (very good) feature characterisation methods, but they are all based on the Gaussian scale space. As an alternative to the Gaussian pyramid, we use the Dual-tree Complex Wavelet Transform, which has similar computational complexity but better directional selectivity, as the basis for our work. The second problem relates to ways of testing the performance of feature characterisation methods. In order to create a good feature characterisation method, one needs a reliable and informative way of testing it. Once we have this, we can find and overcome weaknesses in the method and come up with a reliable overall solution to the problem. Hence we created a fully calibrated dataset specifically for the purpose of automatically evaluating keypoint methods in a full 3D environment. The main contributions of this work are:

BTK: A new wavelet based keypoint detector and descriptor

Although wavelets have proven very successful for image compression, image coding, denoising and deconvolution, there has been little work on using them for local feature based image matching. Similarly, although some phase-based approaches do exist, ([Carneiro and Jepson, 2007], for example), most of the existing work on multiscale keypoint detection is based on conventional real representations such as Difference of Gaussian decompositions [Lowe, 1999, Mikolajczyk et al., 2005], rather than wavelets or complex representations.

Here, we develop an approach to multiscale keypoint matching based on the Dual-Tree Complex Wavelet Transform [DTCWT]. Our approach improves upon its predecessors [Fauqueur et al., 2006, Kingsbury, 2006] in the following ways:

- Our method is based on a densely sampled scale-space pyramid called the 4S-DTCWT that achieves a major reduction in scale-related aliasing.
- We present a new keypoint detector (BTK) based on the 4S-DTCWT. This has improved spatial localisation and better scale estimation owing to the use of a better cornerness measure and keypoint localisation in individual levels.
- The denser sampling in scale in the scale-space pyramid and the improved scale estimates from the detector lead directly to more robust visual descriptions. We also present some ways of speeding up both the descriptor and the matching computation to make it more suited to use in practical settings.

Automatic 3D evaluation of keypoint detectors and descriptors

We develop a novel, fully automated framework for evaluating keypoint detectors and descriptors. This has the following advantages over previous evaluations:

- It is based on a new dataset containing 3978 calibrated images from two cameras of 39 different toy cars on a turn-table. The dataset, the

calibration images, inter-camera calibration, rotational calibration and the MATLAB test scripts are available at http://www-sigproc.eng.cam.ac.uk/imu.

- To the extent possible, we separate the evaluation of the keypoint detectors from that of the descriptors. We also show that, conditioned on an associated keypoint being detected at a reasonably correct corresponding location, the descriptor similarity is hardly affected by viewpoint variation. On the contrary, the repeatability of most keypoint detectors falls off rapidly with changes in viewpoint.

- Our method of establishing ground truth correspondences differs from conventional epipolar line search in that it is less dependent on angular separation between the reference view and the test view because we use a three-image setup. It is also independent of any descriptor because we use normalised cross correlation to establish correspondences between the reference view and an auxiliary view and they have a fixed angular separation throughout the evaluation.

- Finally, we present an enhanced epipolar matching constraint that is useful for eliminating the uncertainties involved in conventional epipolar line search based evaluations.

The development of the keypoint detector and descriptor is described in Chapters 3–4 and the evaluation work is described in Chapters 5–6.

1.3 Outline

In **Chapter 2**, we review prior work on feature detectors and descriptors. Then we describe the Dual Tree Complex Wavelet Transform, explain its applicability to local feature extraction and review existing work in this area. Finally, we provide an overview of the methods used to evaluate feature detectors and descriptors on 2D and 3D scenes. We identify the areas in which more work is needed and use this to guide us through the remainder of the thesis.

In **Chapter 3**, we explain the concept of scale and describe the motivation for formation of our 4S-DTCWT scale-space. This is followed by a discussion of the problems of and solutions for sampling in scale while keeping the computation practical and efficient. We then introduce our new keypoint strength measure and detail our 2D keypoint detection technique that is based on this and the 4S-DTCWT scale-space. The remainder of the chapter explains different methods for estimating the scale of the keypoint, presents a qualitative evaluation of the methods discussed and finally, presents an overview of the preferred configuration of our keypoint detector.

Chapter 4 describes our keypoint descriptor in detail and suggests ways of improving the efficiency of the descriptor and descriptor matching computations.

Chapter 5 presents the Cambridge toy cars dataset, a new 3D dataset that we created to facilitate appearance-based evaluation of keypoint detectors and descriptors. We detail the geometry of the setup and the test framework, followed by a quantitative evaluation of the repeatability of our keypoint detector and descriptor alongside a selection of competing methods. We also present quantitative results for some of the important configuration decisions regarding the keypoint detector, as discussed in Chapter 3.

In **Chapter 6**, we present an alternative method for evaluating keypoint detectors. This geometric method differs from the method used in Chapter 5 in the sense that it incorporates constraints on keypoint scale as well as spatial position. We explain the problem and sketch the theory briefly, followed by a description of the experiments and results.

Finally, in **Chapter 7**, we draw conclusions and discuss possible areas for future work.

Chapters 3, 4, 5 and 6 describe the author's work, except where stated. Appendices B, D and E.2 describe collaborative work. The contributions of the authors are listed in footnotes in each appendix. Appendix A has been included for completeness. Appendices C and E.1 provide helpful implementation details.

Chapter 2

Prior work

In this chapter, we review the state-of-the-art in keypoint detection, description and matching. We also review previous work on evaluating keypoint methods and explain the limitations of these methods.

2.1 Feature extraction techniques

Points, lines or regions in an image that are sufficiently different from their neighbourhoods are called key features [Marr, 1982]. They may or may not, by themselves, give a complete visual description of an object [Marr, 1982], but ideally the key features describe the object sufficiently for it to be distinguished from other objects. Feature detection is used in the early stages of most computer vision tasks. It is usually followed by the use of some higher level heuristics or application-specific knowledge in order to make meaningful inference. Here we review a selection of local feature detection and description methods that represent the current state-of-the-art. This is by no means an exhaustive list of the early-vision techniques used in object recognition, however, it will suffice for the discussion of our work.

2.1.1 SIFT detector and descriptor

Scale Invariant Feature Transform [SIFT] [Lowe, 1999, 2004] is a robust interest point detection, description and matching scheme. It is invariant to

translation and rotation and handles some degree of affine variation as well. The detector finds *blob-like* structures in the images that have a clear inside and outside and that hence tend to be well-localised in both position and scale.

Difference-of-Gaussian detector

The idea of using a scale-space representation for the analysis of images is due to [Marr and Hildreth, 1980] and [Witkin, 1983]. In [Koenderink, 1984] (also later in [Babaud et al., 1986] and [Lindeberg, 1994]), it was shown that, under the constraint that no extraneous detail is generated as the resolution is decreased, a Gaussian kernel is the only one-parameter (resolution being the parameter) solution to the problem of creating a viable scale-space. SIFT builds on these findings.

The scale-space representation of an image is formed by progressively smoothing the image with a Gaussian kernel and using the difference of these successive levels (Difference-of-Gaussian) to efficiently approximate the Laplacian-of-Gaussian[1] function. The image is also decimated by a factor of two every time the width of the Gaussian kernel changes by a factor of two (*i.e.* at each octave). If $I(x, y)$ is the image intensity at location (x, y) in image I and $G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} \exp\{-(x^2+y^2)/2\sigma^2\}$ is the Gaussian smoothing kernel of width σ, then in the scale space representation, the smoothed image at level σ, $L(x, y, \sigma)$ is given by,

$$L(x, y, \sigma) \quad = \quad G(x, y, \sigma) * I(x, y) \tag{2.1}$$

where $*$ represents the convolution operator. The difference-of-Gaussian filtration of the image is given by,

$$D(x, y, \sigma) \quad = \quad L(x, y, k\,\sigma) - L(x, y, \sigma) \tag{2.2}$$

Note that in SIFT the input image is upsampled by a factor of two using linear interpolation before creating the finest level of the scale-space. Local

[1] This approximation is best when the ratio of the width of the two Gaussians is equal to 1.6 as shown in [Marr and Hildreth, 1980] and [Marr, 1982] (pp 62–63).

extrema are extracted at each level separately. The scale-space extrema (maxima and minima) detected in the DoG levels are labeled as potential interest points.

Keypoint localisation

Next, each extremum is validated and refined by performing a detailed fit with its neighbours in the 3D scale-space using a quadratic Taylor expansion of the scale-space function $D(\mathbf{x})$ centred on the potential interest point $\mathbf{x} = (x, y, \sigma)^\top$,

$$D(\mathbf{x}) \;=\; D + \frac{\partial D^\top}{\partial \mathbf{x}}\,\mathbf{x} + \frac{1}{2}\,\mathbf{x}^\top\,\frac{\partial^2 D}{\partial \mathbf{x}^2}\,\mathbf{x} \quad . \tag{2.3}$$

The location of the quadratic extremum, $\hat{\mathbf{x}}$, is estimated as

$$\hat{\mathbf{x}} \;=\; -\left(\frac{\partial^2 D}{\partial \mathbf{x}^2}\right)^{-1}\,\frac{\partial D}{\partial \mathbf{x}} \quad . \tag{2.4}$$

A keypoint is retained if the value of the scale-space function at the extremum

$$D(\hat{\mathbf{x}}) \;=\; D + \frac{1}{2}\,\frac{\partial D^\top}{\partial \mathbf{x}}\,\hat{\mathbf{x}} \tag{2.5}$$

satisfies the condition $|D(\hat{\mathbf{x}})| > 0.03$ (*i.e.* has a sufficiently good contrast with respect to the neighbourhood), otherwise it is rejected.

SIFT rejects unstable keypoints located on edges [Brown and Lowe, 2002]. To do this, the 2×2 Hessian matrix $\mathbf{H}$ is computed at all the candidate keypoint locations, $\mathbf{x}$, using second derivatives (D_{xx} denotes the second derivative of D in the direction x)

$$\mathbf{H}(\mathbf{x}) \;=\; \begin{bmatrix} D_{xx}(\mathbf{x}) & D_{xy}(\mathbf{x}) \\ D_{xy}(\mathbf{x}) & D_{yy}(\mathbf{x}) \end{bmatrix} \quad . \tag{2.6}$$

Then, the following criterion is tested at all candidate keypoint locations

$$\frac{r}{(1+r)^2} = \frac{\mathrm{Det}(\mathbf{H})}{\mathrm{Tr}^2(\mathbf{H})} \quad \leq \text{Threshold}, \tag{2.7}$$

where $r = \lambda_1/\lambda_2$ is the ratio of eigenvalues of $\mathbf{H}^2$. Keypoints located on edges have large principal curvature in one direction but a small one in the perpendicular direction. All keypoints that do not satisfy criterion in (2.7) are discarded. In practice, a value is set for r, (thus for $r/(1+r)^2$) and the value of $\mathrm{Det}(\mathbf{H})/\mathrm{Tr}^2(\mathbf{H})$ computed from the Hessian is checked against this threshold. This approach avoids having to explicitly compute the individual eigenvalues of the Hessian [Harris and Stephens, 1988]. At this stage, each keypoint is characterised by its location and scale.

Orientation assignment

Gradient magnitude $m(x, y)$ and orientation $\theta(x, y)$ is computed at each sample in the scale-space, $L(x, y)$ for all scales σ as

$$m(x,y) = \sqrt{(L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))^2} \tag{2.8}$$

and

$$\theta(x,y) = \tan^{-1}\left(\frac{L(x,y+1) - L(x,y-1)}{L(x+1,y) - L(x-1,y)}\right) \quad . \tag{2.9}$$

Although gradient values are only required at and around keypoints, for efficiency reasons, they are pre-computed at all scales and all samples.

Using the observed scale σ of the keypoint, a region with full-width $= 6\,\sigma$ centred at the keypoint within the scale-space level $L(x, y, \sigma)$ is extracted as the neighbourhood of the keypoint. Gradient orientations within this neighbourhood, weighted by a Gaussian weighting of standard deviation 1.5σ and the respective gradient magnitudes are used to form an orientation histogram. A separate keypoint is stored for each significant peak in this orientation histogram. The resulting keypoints are characterised by their location, scale and orientation.

SIFT descriptor

To form a descriptor, pre-computed gradient magnitudes and orientations

[2]The quantity $r/(1+r)^2$ is invariant to the exact order of the eigenvalues, r can be replaced by $1/r$ without changing the value of $r/(1+r)^2$

in eight directions are sampled over a 16×16 sample array centred on the keypoint location at the level closest to the keypoint scale. A Gaussian weighting with σ equal to half the descriptor window, and centred over the keypoint, is used to weight the gradients over a 16×16 sample array. This weighting ensures that the gradients close to the keypoint have a greater effect on the descriptor than those further away from it. The gradients in the descriptor window are accumulated into a set of 4×4 orientation histograms, each of which summarizes the information in a 4×4 sample area. Before doing the accumulation, the 16×16 sampling array and the gradient orientations are rotated such that the patch is aligned with the orientation of the keypoint. The gradient histograms are then arranged into a $4\times4\times8 = 128$-element descriptor vector. This results in a rotation-invariant descriptor. Note that information from only one scale, the one closest to the scale of the keypoint, is used in the descriptor.

In order to avoid boundary effects, trilinear interpolation is used to distribute the contribution of each gradient sample into adjacent histogram bins. In order to ensure some robustness to illumination variations, the descriptor is normalised to unit length. If the result has any entries greater than 0.2, then these are clipped at 0.2 and the descriptor is normalised again.

The descriptors are compared using the Euclidean distance metric. If a pair of keypoints has ratio of closest neighbour distance to second-closest neighbour distance less than a certain threshold (usually set to 0.6), then it is considered to be a match, otherwise the pair is rejected. An efficient method for nearest neighbour search of the descriptors in the 128-dimensional space has been proposed in [Beis and Lowe, 1997].

Several modifications of the original SIFT algorithm have been proposed. A few notable ones are PCA-SIFT [Ke and Sukthankar, 2004] (PCA based dimensionality reduction of the SIFT descriptor), the Speeded Up Robust Feature (SURF) descriptor [Bay et al., 2006, 2008] (based on *integral images* [Viola and Jones, 2001]), and Global SIFT for augmenting location with the SIFT descriptor's spatial information [Mortensen et al., 2005]. An open-source implementation of the SIFT detector and descriptor is [Vedaldi and Fulkerson, 2008]. An elaborate description of feature detection and scale

selection methods and the associated scale-space theory for continuous signals can be found in [Lindeberg, 1998].

2.1.2 Harris-Affine & Hessian-Affine detectors

Affine-covariant detectors detect regions of elliptical or arbitrary shape whose shape is intended to adapt to the underlying image transformations. The Harris-Affine and Hessian-Affine detectors [Mikolajczyk and Schmid, 2004] are two such methods that use scale-space interest points as their starting point.

For an image I, let $\mathbf{x}$ be the location vector of a keypoint, and let I_x and I_{xx} denote respectively the first and second derivative of I in the direction x. Let $G(\sigma_I)$ denote a Gaussian window of width σ_I and let σ_D denote the differentiation scale for the Difference-of-Gaussian operation. The Harris-Affine [HAR–AFF] detector is based on the Harris corner detector [Harris and Stephens, 1988]. The second moment matrix,

$$\mathbf{M} = \sigma_D^2 G(\sigma_I) * \begin{bmatrix} I_x^2(\mathbf{x}, \sigma_D) & I_x(\mathbf{x}, \sigma_D) I_y(\mathbf{x}, \sigma_D) \\ I_x(\mathbf{x}, \sigma_D) I_y(\mathbf{x}, \sigma_D) & I_y^2(\mathbf{x}, \sigma_D) \end{bmatrix} \quad (2.10)$$

which is based on the squared gradients of the image is used to locate interest points and subsequently to estimate the shape and extent of the interest point region [Mikolajczyk and Schmid, 2004, Baumberg, 2000, Lindeberg, 1995].

The Hessian-Affine detector is based on the Hessian detector [Beaudet, 1978]. It uses the Hessian matrix, composed of second derivatives

$$\mathbf{H} = \begin{bmatrix} I_{xx}(\mathbf{x}, \sigma_D) & I_{xy}(\mathbf{x}, \sigma_D) \\ I_{xy}(\mathbf{x}, \sigma_D) & I_{yy}(\mathbf{x}, \sigma_D) \end{bmatrix} \quad (2.11)$$

to locate interest points and to estimate their shape. The Hessian-Affine [HES–AFF] detector mostly picks up blobs whereas the Harris-Affine detector mostly picks up points of high local curvature and highly textured regions. For scale estimation, both detectors seek maxima of the Laplacian response at the location of the interest point over a range of scales. Finally, the square

root of the second moment matrix or Hessian matrix is used to estimate the shape of the region around the interest point. Using the estimated shape, the region is normalised into a circular one by a local affine warping. The location and scale of the interest point are re-detected (refined) over the normalised patch. The process is repeated until the eigenvalues of the second moment matrix for the normalised patch are equal *i.e.* the differential warping is an identity matrix. If $\mathbf{x}_1$ and $\mathbf{x}_2$ are corresponding patches detected in two different views of a scene, such that they are related by an affine transformation, then each of these patches is transformed to a normalised version by the corresponding second moment matrix as

$$\mathbf{x}'_1 = \mathbf{M}_1^{-1/2}\,\mathbf{x}_1 \quad \text{and} \quad \mathbf{x}'_2 = \mathbf{M}_2^{-1/2}\,\mathbf{x}_2 \tag{2.12}$$

$$\text{so that as a result} \quad \mathbf{x}'_2 = \mathbf{R}\,\mathbf{x}'_1 \tag{2.13}$$

i.e. the normalised versions $\mathbf{x}'_1$ and $\mathbf{x}'_2$ are related by a simple rotation $\mathbf{R}$. Thus the inputs to the descriptor calculation are affine-normalised image patches having only rotational ambiguities. Assuming that the descriptor calculation removes these ambiguities, the result is a keypoint detection and description mechanism that is invariant to affine deformations of the image and hence moderate changes of viewpoint.

2.1.3 Maximally stable extremal regions

Maximally stable extremal regions were introduced in the context of wide baseline stereo correspondence [Matas et al., 2002]. This feature detector aims to detect image regions that are highly resistant to illumination variations. A series of binary images is generated by thresholding the input image. Every grey level present in the image is used as a threshold to produce one image. Regions are found in each of these images and those regions which exhibit least variation in area across a range of successive thresholds are marked as *maximally stable extremal regions* [MSER]. Rather than explicitly calculating a series of thresholded images, it makes use of an efficient implementation of the watershed algorithm [Vincent and Soille, 1991], that

has a complexity of $O(n \log \log n)$. This strategy identifies regions enclosed by sharp changes in intensity. Variants find light regions enclosed by dark regions and vice versa. The minimum number of grey level thresholds over which a region needs to persist is called the margin of the MSER detector and is an important variable parameter [Matas et al., 2002]. Owing to the threshold-based extraction method, these regions are tolerant to affine variations. For affine-invariant description, the detected regions are normalised so as to have unit eigenvalues by affine warping. A SIFT descriptor is used to describe the normalised region. Further details can be found in [Matas et al., 2002].

This approach is capable of handling a great deal of affine variation and has proven to be good for establishing correspondences due to its high repeatability. It performs very well on strongly contrasted planar regions without much texture such as lettering. Usually, it performs less well on natural scene regions containing rich multiscale texture or noise.

2.1.4 Edge-based regions & Intensity based regions

The edge based region [EBR] detector [Tuytelaars and Van Gool, 2004] uses a Harris corner point and a pair of edges centred on the corner as pivots to create an affine-covariant region. Edges can be extracted stably over a range of viewpoints and illumination variation. Beginning at the pivot point, the two edges are tracked simultaneously such that their relative speed is coupled via affine invariant parameters, leading to a family of parallelograms with one vertex and two edges fixed at the pivot. A further criterion using the photometric quantities of the region is applied to select one or a few of these parallelograms as interesting regions. A region is declared to be located at the centre of gravity of its parallelogram and its extent is determined by the parallelogram.

The intensity extremum based region [IBR] detector [Tuytelaars and Van Gool, 2004] starts from an intensity extremum and evaluates an intensity

function of the form

$$f_I(t) = \frac{\text{abs}(I(t) - I_0)}{\max(\frac{\int_0^t \text{abs}(I(t)-I_0)dt}{t}, d)} \tag{2.14}$$

along several rays emanating from the point. Here t is the distance along the ray, $I(t)$ is the intensity at distance t, I_0 is the intensity at the seed point (intensity extremum) and d is used to avoid division by zero. The method then chooses a location extremum of $f_I(t)$ along each ray (these are typically locations at which the intensity changes abruptly along the ray) and the set of all such points delineates a region of arbitrary shape. For affine-covariant behaviour and simplicity, this region is then approximated by an ellipse centred on the intensity extremum with the same second moment matrix as the intensity based region. Intensity extrema are detected at multiple scales to make the detector multiscale. Further details of both methods can be found in [Tuytelaars and Van Gool, 2004].

2.1.5 DAISY descriptor

The *DAISY* descriptor configuration has been shown to work well in several local image descriptor studies *e.g.* [Hua et al., 2007, Winder and Brown, 2007, Winder et al., 2009, Brown et al., 2011]. The basic idea is to pool oriented gradients over circular regions arranged in concentric rings around the interest point and to machine-optimise the descriptor parameters for optimal matching performance. The DAISY framework comprises three essential stages of processing followed by two optional ones. The first block consists of the formation of a fixed length feature vector for every pixel in the region being described (usually a scale normalised canonical patch using filter-bank based gradient computations). The second block accumulates the feature vectors into spatial bins using a Gaussian weighting. The bins are circular and arranged around the central point in a log-polar arrangement (similar to GLOH [Mikolajczyk and Schmid, 2005], Geometric Blur [Berg and Malik, 2001] and Shape Context [Belongie et al., 2000, 2002]). The bins get larger as one moves away from the centre. The bins in adjacent rings are

offset by half the angular bin width. There can be variable numbers of rings and within each ring there can be variable numbers of bins. The numbers of rings and bins per ring are parameters that are machine-optimised for a particular task (by maximising area under the relevant ROC curve[3]). The normalisation block uses geometric length normalisation of the combined feature vector followed by a high dynamic range compression by clipping the values in the descriptor to a pre-determined value. This is followed by an optional Principal Components Analysis block for dimensionality reduction and by an optional quantization and compression block for compact storage. Further details can be found in [Winder et al., 2009], [Brown et al., 2011] and information on the efficient computation of DAISY descriptors can be found in [Tola et al., 2008].

2.1.6 Efficient implementations

A number of simplified and/or efficient implementations of keypoint detectors and descriptors exist for specific applications. A few of the most notable ones are:

- Features from Accelerated Segment Test (FAST): including the FAST detector [Rosten and Drummond, 2005, 2006], a FASTer version [Rosten et al., 2010] and its predecessor the SUSAN detector [Smith and Brady, 1997]
- A fast keypoint recognition system for wide-baseline matching using trained model images [Lepetit and Fua, 2006]
- Speeded Up Robust Features (SURF) [Bay et al., 2008]
- Fast SIFT-like descriptors [Tola et al., 2008]

The goal of these fast detectors and descriptors is slightly different from the ones considered in this thesis. Methods focussing on efficiency often approximate conventional feature detectors in some respect, typically being tuned

[3]**ROC curve**: Receiver Operating Characteristic (ROC) curve plots the true positive rate against the false positive rate for a particular parameter setting in an algorithm. A ROC curve is obtained by varying the parameter over the entire range of values.

for the processing of large amounts of data in real time, often with minimal resources, for example video streams. This thesis deals with conventional feature detectors that aim to find highly repeatable and robust features in still images with efficiency being of secondary importance for now. Of course, any feature detector/descriptor can also benefit from platform specific implementations [Sinha et al., 2006, Heymann et al., 2007, Cabani and MacLean, 2007].

2.1.7 Discussion

Having reviewed some of the leading contemporary feature detection and description methods, it is clear that feature characterisation methods mostly fall into two broad categories. Either they are interest point detectors based on repeated smoothing of an image with a Gaussian filter, or they are region detectors exploiting luminance patterns, or in some cases they employ a combination of both. But, there is, to our knowledge, no stable interest point detector that is not based on the Gaussian scale space. There is ample evidence from physiological experiments that the brain possesses orientation sensitivity that is at least as good as 30° [Hubel et al., 1977, Hubel, 1995]. Computer-based recognition systems may also perform better if they possess similar or better directional sensitivity. Derivatives of Gaussians provide a directional resolution of 90° (between -3dB points, *c.f.* Figure 2.1-a), but there exist other decomposition methods that provide multi-scale gradients at finer angular resolution, such as Steerable Pyramid [Simoncelli and Freeman, 1995] and the Dual-Tree Complex Wavelet Transform [Kingsbury, 2001]. We shall be focussing on developing an interest point detector and an associated descriptor using one of these tools (DTCWT), in the hope of achieving more directionally sensitive feature characterisation.

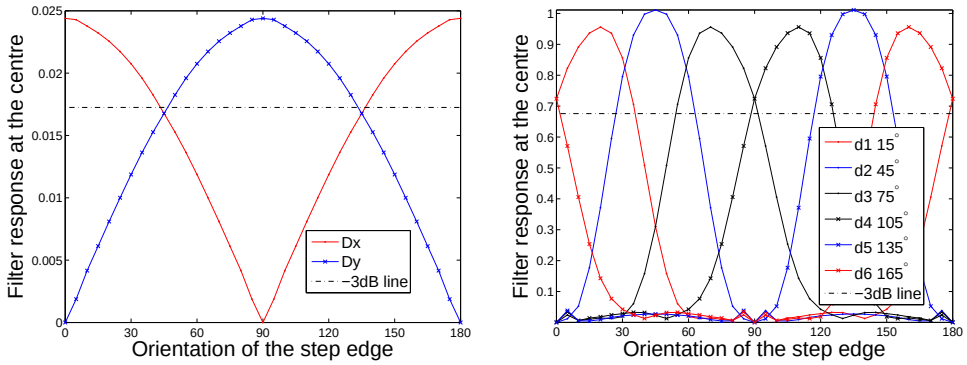
2.2 Dual-Tree Complex Wavelet methods

In this section we give an overview of the Dual-Tree Complex Wavelet Transform (DTCWT) and existing work on keypoint detection and description that

uses the DTCWT.

2.2.1 Dual-Tree Complex Wavelet Transform

The Dual-Tree Complex Wavelet Transform [DTCWT] of a 1D signal uses two carefully designed dyadic trees to compute the real and imaginary components of a complex analytic wavelet decomposition using only efficient real arithmetic. The filters in the two trees are all real and Hilbert Transform pairs of each other. In 2D, the DTCWT is designed [Kingsbury, 2001] to output six analytic and directionally sensitive subbands oriented at $(30d - 15)^\circ$ for $d = 1 \ldots 6$ (see Figure 2.1-b).



(a) Gaussian derivatives (b) DTCWT filters

Figure 2.1: Derivatives of Gaussian (*left*) and DTCWT filter responses (*right*) for a step edge input whose orientation varies from 0° to 180° in steps of 5°. The response in the DTCWT subbands varies smoothly in accordance with the orientation of the input step edge. The filter width (measured at -3dB points) for DTCWT filters is about 35° and that for derivatives of Gaussians is about 90°. DTCWT filters are thus directionally more selective than derivatives of Gaussians.

The DTCWT uses *fixed rectangular* partitioning of the frequency plane. This allows a linearly separable filter design that has a linear phase response while allowing perfect reconstruction. DTCWT has a redundancy of 4 : 1 for images. In comparison, the Steerable Pyramid [Simoncelli and Freeman, 1995] (based on the steerable filters [Freeman and Adelson, 1991]) has a *polar-steerable* partitioning of the frequency plane. This has the benefit of

being able to choose the number of orientations but the filters are not linearly separable. The output of the steerable pyramid has a redundancy of $8m/3$ for (shift-invariant) complex filters and $4m/3$ for (shift-dependent) real filters where m is the number of orientations. Thus, the main attraction of the DTCWT is that it is a reversible, energy-preserving, wavelet transform that is analytic yet has a separable filter bank implementation at a limited redundancy.

A more detailed mathematical analysis of the Dual Tree Complex Wavelet Transform (DTCWT) can be found in [Selesnick et al., 2005]. Design of the DTCWT, its properties and related filter design issues are discussed in [Kingsbury, 1999].

DTCWT is particularly suitable for our application because it is approximately shift-invariant, directionally selective and has a separable and hence efficient implementation.

The wavelet coefficients (band-pass outputs) are denoted by $H_k(x, y, d)$ and the scaling coefficients (low-pass outputs) by $L_k(x, y)$, where k is the DTCWT level, d is the subband direction and (x, y) are the spatial variables. k takes values $(1, \ldots, N)$ for an N level DTCWT and d takes values $(1, \ldots, 6)$.

2.2.2 DTCWT with improved rotational symmetry

The rotational symmetry of the DTCWT can be further improved (*c.f.* [Kingsbury, 2006]) by adding a bandpass filter in each direction and doing a phase correction to make the responses conjugate symmetric. This provides significantly better shift invariance and orientation selectivity than conventional real discrete wavelet transforms (DWTs) at lower computational cost than a comparable steerable filter.

Although the DTCWT as described in [Kingsbury, 2001] has attractive perfect reconstruction properties, it is not rotationally symmetric. For feature description, the perfect reconstruction constraint can be relaxed to create a more rotationally symmetric version of the DTCWT. The 45° and 135° subband centre frequencies of the DTCWT are further away from the origin than the other four subbands at a given scale in the frequency spectrum.

This is because the centre of the 1D Hi filter is three times further from the origin than the 1D Lo filter (because they both span half the bandwidth of the input signal), so a 2D Lo-Hi filter formed from a combination of 1D Lo and 1D Hi is closer to the origin than a 2D Hi-Hi filter by a factor of $\sqrt{3^2+3^2}$ $/\sqrt{3^2+1^2} = \sqrt{1.8}$.

An additional bandpass filter may be added in each dimension to pull the 45° and 135° subband centre frequencies closer to the origin by $\sqrt{1.8}$ [Kingsbury, 2006].

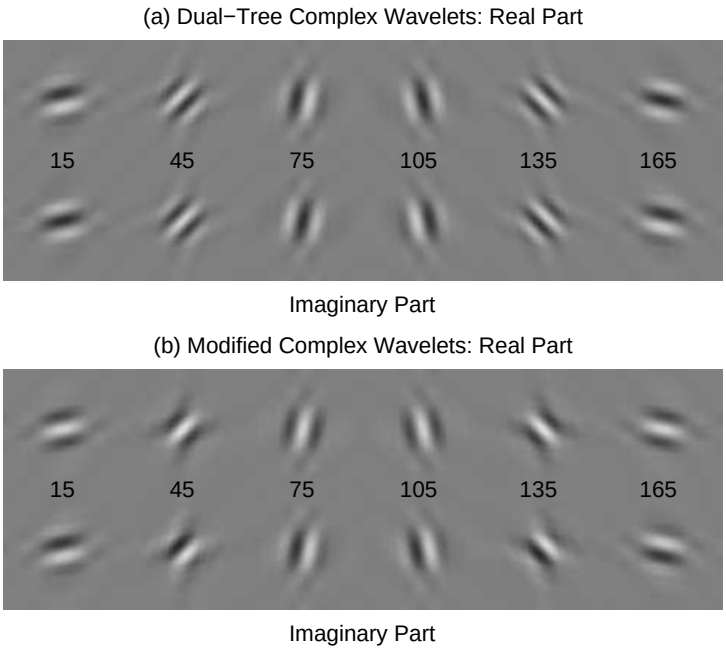


Figure 2.2: Impulse responses of the DTCWT before and after the addition of an extra bandpass filter in each dimension and phase correction to have zero phase at the mid-point of the responses. This results in a more rotationally symmetric DTCWT. Figures reproduced from [Kingsbury, 2006].

Another feature of the standard DTCWT is that all six subbands may not have zero phase at the mid-point of their responses. As described in [Kingsbury, 2006], a phase correction of $\{j, -j, j, -1, 1, -1\}$ is been applied to make all real parts of the six subband responses even symmetric and imaginary parts of all the six subband responses odd-symmetric. This property allows one to calculate responses in the opposing directions $(30d - 15 + 180)°$

by conjugating the responses of the original six subbands, $(30d - 15)°$. The orientation of zero crossing changes in a cyclic manner across the six subbands. The phase-corrected impulse responses are compared in Figure 2.2.

2.2.3 FKA Keypoint Detector

Our work builds on, and significantly improves, an earlier DTCWT based keypoint detector [Fauqueur et al., 2006]. This earlier version (described in this section) detects keypoints at the maxima of the "accumap" – an accumulated map of responses across scale and orientation, $\sum_k E_k(x, y)$, where

$$E_k(x, y) \equiv \prod_{d=1..6} |H_k(x, y, d)|^{1/4} \tag{2.15}$$

and $H_k(x, y, d)$ is the complex DTCWT coefficient at level k, subband (orientation) d and location (x, y). The moduli of the wavelet coefficients characterise the oriented gradient energy at the given position and scale, so taking their product over orientations gives a response reminiscent of a Harris (determinant of oriented energy tensor) detector. Summing these responses over all tree levels provides a degree of scale invariance. Given the (x, y) position of an accumap maximum, the scale of the corresponding keypoint is estimated by searching for the first radial distance at which the sum of outward gradients of the accumap has a strong minimum. The gradients are computed at 8 fixed radial directions at intervals of 45°. This forms the starting point of our work in Chapter 3 and is described in more detail there.

2.2.4 DTCWT Local Descriptor

To complement our detector we use rotation-invariant DTCWT-based "Polar Matching Matrices" as the local visual descriptor [Kingsbury, 2006] for our keypoints. We briefly describe this descriptor.

Polar matching matrix (**P**-matrix) descriptors are created from DTCWT coefficients as follows [Kingsbury, 2006]. At a designated DTCWT level and sampling radius, a circle of 12 points spaced 30° apart is placed around the central point (keypoint), and for each DTCWT orientation, its complex

DTCWT coefficient is evaluated at each point, using spatial interpolation in the DTCWT response as necessary[4]. There are 12 orientations (6 spaced 30° apart and their complex conjugate pairs 180° away from these). If the resulting coefficients are arranged in a 12×6 complex matrix (which comprise the central 6 columns of a standard 12×8 **P**-matrix) with column c ($c = 1 \ldots 6$) containing the coefficients whose orientation relative to the tangent to the sampling circle at the sample direction is $(30c - 15)°$, then rotations by multiples of 30° produce cyclic shifts within each column of the matrix, *i.e.* simple phase changes of the FFT of the column. This property allows efficient rotation-invariant descriptor comparison and efficient estimation of the relative angle between the two descriptors. To produce a complete **P**-matrix descriptor, matrices from several circles with different radii and/or levels can be appended, and additional columns can be included based on the coefficients of the 12 orientations (6 conjugate pairs) at the central point at a given level. The most conventional arrangement [Kingsbury, 2006] is a spatially-compact local descriptor whose 12×8 matrix contains the coefficients from the circle with radius one sample spacing at the given level, the central point at that level, and the central point at the next level up (2× coarser). For illumination invariance, the total energy in each **P**-matrix is normalised, so that matching them produces a correlation score in the range $[-1, 1]$. This forms the basis of our work in Chapter 4 and is described in more detail there.

Other rotation-invariant descriptors include [Schmid and Mohr, 1997, Schaffalitzky and Zisserman, 2002, Mikolajczyk and Schmid, 2005] based on [Koenderink and Van Doorn, 1987] and [Carneiro and Jepson, 2007].

2.3 Evaluation techniques

Several studies have explored different approaches to evaluating keypoint detectors and descriptors. Some have concentrated on scale changes [Mikolajczyk and Schmid, 2001, Lindeberg, 1998], while others have concentrated

[4]Such interpolation is reliable owing to the band-limited nature of the rotationally symmetric DTCWT.

more on affine variations [Mikolajczyk et al., 2005, Mikolajczyk and Schmid, 2005]. Yet others have tested the tolerance of a combination of image transformations [Schmid and Mohr, 1997, Lowe, 2004]. Detailed studies on evaluation techniques include [Mikolajczyk, 2002, Brown, 2005]. In the following sections, we describe these recent studies that are most relevant to our work.

2.3.1 Evaluation of keypoints on planar scenes

A comprehensive quantitative evaluation of keypoint detectors and descriptors was performed by [Mikolajczyk et al., 2005] and [Mikolajczyk and Schmid, 2005]. This framework was developed for *affine-invariant* (strictly, affine-covariant) region detectors and descriptors. It is based on images of planar scenes and deformations of images taken from a fixed camera position so that all of the images in a set are related by planar homographies. (The depth of scene is small compared to the distance from the camera.)

The dataset contains 6 sequences of photographs with viewpoint changes between 0° and 70° and 10 sequences containing scale changes by factors of 1.4 to 4.5. Some sequences also have degradations like rotation, image blur, JPEG compression, or illumination changes. The images have a resolution of approximately 800×640 pixels, but their sizes vary within the dataset. The ground truth homographies between pairs of corresponding images is computed using a two step process. First an approximate homography is estimated using manually selected corresponding points. Then the image is warped using the approximate homography, interest points are detected and matched automatically in the warped image and the original image. These correspondences and the approximate homography are used to estimate a final homography.

The performance of the region detectors is measured by the repeatability criterion

$$\epsilon_S \;=\; 1 - \frac{|\mu_a \cap H(\mu_b)|}{|\mu_a \cup H(\mu_b)|} \tag{2.16}$$

where μ_a and μ_b are the two (usually elliptic) regions being compared. Here,

$H(\mu_b)$ is the projection of the region μ_b from image B onto image A and images A and B are related by a homography[5] H. The operator $\cup$ denotes the point-wise union and $\cap$ denotes the point-wise intersection of the two regions and $|.|$ denotes the surface area of the result. The overlap error ϵ_S varies between unity for no overlap and zero for complete overlap. The region is accepted as a valid match if $\epsilon_S < 0.4$.

The descriptors are evaluated by examining the slope of *precision-recall*[6] curves (recall versus 1-precision graphs) where

$$recall = \frac{\#correct\ matches}{\#correspondences} \quad \text{and} \tag{2.17}$$

$$1 - precision = \frac{\#false\ matches}{\#correct\ matches + \#false\ matches} \tag{2.18}$$

The dataset and its associated evaluation framework are available for download from [Mikolajczyk, 2005]. The availability of this dataset and evaluation framework, its ease of use and good documentation has allowed many detectors and descriptors to be tested against their benchmark. This study has more or less become *the* standard in evaluation of keypoint detectors/descriptors on planar scenes.

2.3.2 Evaluation of keypoints on non-planar scenes

Evaluations of keypoint detectors and descriptors on 3D scenes include [Fraundorfer and Bischof, 2005] and [Moreels and Perona, 2005, 2007]. The ground truth is established using purely geometric constraints in both these methods, unlike [Mikolajczyk et al., 2005] which uses appearance as well as geometry.

The evaluation in [Fraundorfer and Bischof, 2005] uses trifocal tensors (see Appendix A for an explanation of a trifocal tensor) to estimate the

[5] If the ellipses corresponding to the μ_a and μ_b are written in their 3×3 matrix representation, then $H(\mu_b) = H^\top \mu_b H$.

[6] Precision-recall (PR) curve is a closely related alternative representation of the ROC curve. A PR curve plots the true positive rate against the ratio of true positives to all positives, and is usually the preferred representation in case of highly skewed datasets, as it shows the weaknesses of an algorithm more clearly than a ROC curve [Davis and Goadrich, 2006], [Szeliski, 2010].

ground truth correspondences on one office scene and one sequence of two boxes on a turn-table. Both sequences are composed of both planar regions and 3D objects with significant depth discontinuities. The sequences contain images taken at 5° intervals over a range of 90°. This study evaluates the performance of keypoint detectors on planar and non-planar scenes separately but the evaluation method is the same as in [Mikolajczyk et al., 2005] and the number of images is rather limited[7].

Another recent evaluation of keypoint detectors and descriptors on 3D scenes using calibrated images on a turn-table is [Moreels and Perona, 2005, 2007]. This study combines the evaluation of detectors and descriptors, using ground truth estimated from the geometry as a means of verifying appearance matches. The evaluation protocol uses three images, a reference image, an auxiliary image and a test image. The corresponding location of a point detected in the reference image is found in the auxiliary image by seeking the best match (according to descriptor matching score) within a certain distance from the epipolar line projected into the auxiliary image. The possible location of the point in the test image is then determined from the intersection of the epipolar lines of the correspondence from the auxiliary and the reference images. Finally, a detector-descriptor pair is declared to match if there is an appearance match for any detected point that lies within a certain distance from the estimated location. If no correspondence is found in the auxiliary image, the reference point is discarded and not used in any tests. The dataset contains images of 100 3D objects on a turn-table. Some of these sequences also contain scale and/or illumination changes. More details on this dataset and its use for the evaluation of keypoint detectors and descriptors can be found in Chapter 5.

While this study performs a comprehensive 3D evaluation of keypoint detectors and descriptors, the associated ground truth is not straight-forward to use. There are 13 or more different sets of calibration images and non-trivial knowledge of a calibration toolbox is needed to compute the information (*i.e.*

[7]There are two sequences containing 19 images each. This dataset and the ground truth correspondences became publicly available shortly after we finished making our Cambridge toy cars dataset, which is described in Chapter 5

the Fundamental matrix that relates any two views of an object) required in the evaluation process[8]. Such challenges motivated us to create a new 3D dataset, develop a generic evaluation framework and make the ground truth (Fundamental matrices) available to facilitate further research in this area. We describe our evaluation framework and the evaluation based on it in Chapter 5.

2.4 Summary

Although wavelets have proven very successful for image compression, image coding, denoising and deconvolution, there has been little work on using them for local descriptor based image matching. Similarly, although some phase-based approaches do exist, *e.g.* [Carneiro and Jepson, 2007], most of the existing work on multiscale keypoint detection is based on conventional real representations such as Difference of Gaussian decompositions [Lowe, 1999, Mikolajczyk et al., 2005], not on wavelets or complex representations. DTCWT based local features have previously appeared in [Fauqueur et al., 2006] and [Kingsbury, 2006] with some work on evaluation.

In this thesis, we develop an approach to multiscale keypoint matching based on the critically sampled Dual-Tree Complex Wavelet pyramid. We also do a thorough evaluation of our method alongside other popular local feature methods. For this purpose, we created a new 3D dataset and evaluation framework, learning from the problems in earlier works on evaluation. Finally, we consider geometric methods for evaluating multiscale keypoint detectors.

[8]This is noted on the web site for the dataset http://www.vision.caltech.edu/pmoreels/Datasets/TurntableObjects/README.txt.

Chapter 3

BTK keypoint detector

In this chapter, we describe a new keypoint detector based on the DTCWT. This detector was introduced in [Bendale et al., 2010b] and the detector is named after the authors (BTK for Bendale - Triggs - Kingsbury). We discuss several approaches for each stage of the keypoint detector and test the performance of these in the context of image matching. A more detailed experimental evaluation of the final detector is presented in Chapter 5.

3.1 Scale

We begin by defining the terms *scale* and *level* in the context of keypoint detection. *Scale* refers to the measure of the size of a feature in an image. Scale is measured in pixels, but it may take sub-pixel values *i.e.* it is a *continuous* variable. In practical scale-space representations, images are typically represented at a set of fixed scale factors, called *levels*, at which there is a direct mapping between the *scale* and the *level*. Each level corresponds to a unique discrete scaling of the input image. Conversely, image features can have any continuous scale and the accuracy of measurement of their scales should be limited only by the image dimensions and numerical precision. On the other hand, the keypoint responses, gradient magnitude information *etc.* are available only at quantized scale values corresponding to the levels. In the ideal case, the system would have as many levels as there can be scales, but in practice, computation is limited so there are only a finite number of

levels over the complete range of possible scale values.

3.2 Sampling in scale: The 4S-DTCWT

Our basic goal is to develop a DTCWT based detector that does not fire inappropriately on edges (as illustrated in Figure 3.6) and that provides accurate subpixel keypoint position and scale estimates that transform appropriately under small translations and dilations (spatial scalings) of the input signal. Regarding the second point, for a minimally sampled filter, the DTCWT already provides exceptionally well modulated responses to translations owing to its complex analytic design, so no improvement is needed here beyond accurate subpixel feature localisation. Unfortunately, the same can not be said of scale resilience: although they are complete as a representation, dyadic (power-of-two) wavelets are too coarsely sampled in scale to prevent the aliasing of energy between levels under small dilations of the input signal, and this makes dyadic scale estimation intrinsically unreliable.

To remedy this our method uses several DTCWT trees (typically 4 trees suffice) rather than just one, interleaving them in scale to provide denser scale sampling and using an appropriate scale-normalisation within each DTCWT tree.

3.2.1 Scale normalisation

For invertibility the original DTCWT uses energy-preserving wavelet normalisation, whereas image resampling typically aims to preserve the range of grey-level values of the local signal, not its energy. We adopt grey-level based normalisation to ensure that keypoint responses, detection thresholds *etc.*, are independent of scale. Hence we scale down the level-k DTCWT wavelet coefficients by 2^{-k} (*i.e.* $2^{-k/2}$ for each of the 2 dimensions of the image). If $H_k(x, y, d)$ is the wavelet coefficient at level k and location (x, y) in orientation d, then the scale-normalised wavelet coefficient $\overline{H}_k(x, y, d)$ is given by

$$\overline{H}_k(x, y, d) \;=\; 2^{-k}\, H_k(x, y, d) \quad \text{for } k = 1 \ldots K. \tag{3.1}$$

In a pyramidal scale-space, there are $W/2^k \times H/2^k \times 6$ DTCWT coefficients at level k where W and H are the image width and height respectively.

3.2.2 Interleaved DTCWT trees

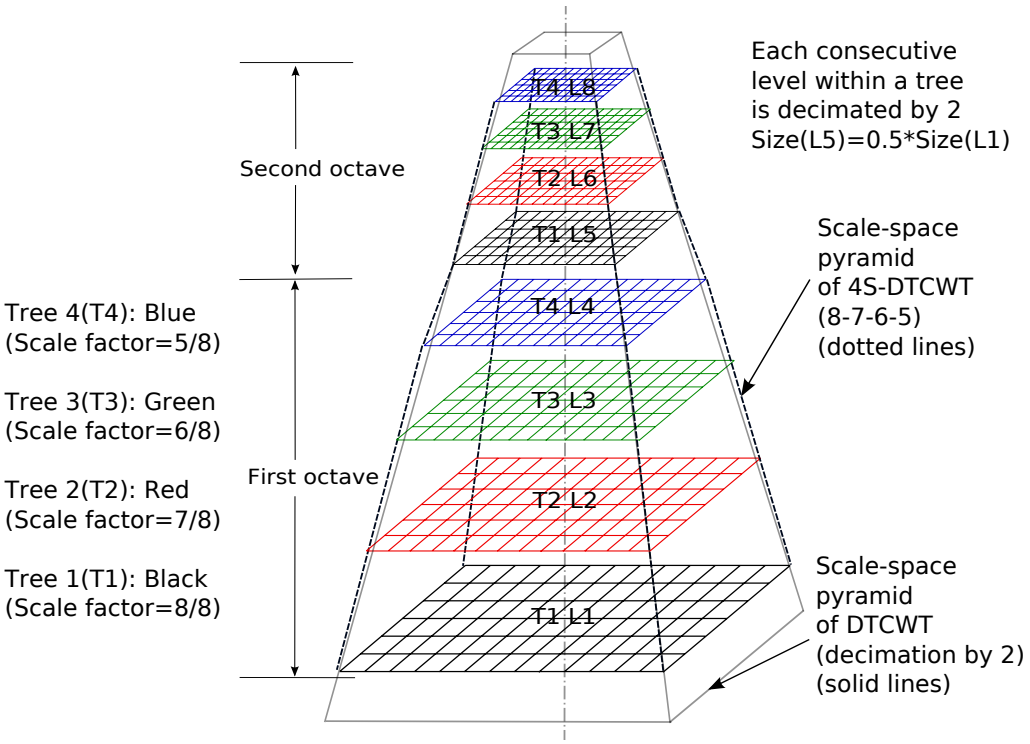


Figure 3.1: Construction of the 4S-DTCWT using the DTCWT and scaled images

Ideally, consecutive levels of a 4-tree pyramid would be spaced at scale intervals of $2^{-1/4}$, but for implementation reasons it is easier to space them at scales of $[1, \frac{7}{8}, \frac{6}{8}, \frac{5}{8}]$ and experimentally we find 4 trees with these spacings suffice for good results. The method thus scales the input image by these amounts using bilinear interpolation and then evaluates separate DTCWT trees over each replica (as usual, decimating by factors of 2 and padding the result to an even size as needed for the calculation of further levels). If the basic DTCWT of the image has K levels we evaluate $K-1$ levels in each of the three new trees, thus producing a pyramid with $4\,K-3$ levels in all.

The construction of 4S-DTCWT is shown in Figure 3.1.

Note that the scale-normalisation is applied within each DTCWT tree. We will refer to the resulting scale-normalised 4-tree DTCWT as the **4S-DTCWT** and denote the corresponding wavelet coefficients[1] by $\widetilde{H}_k(x, y, d)$. The process of computing the 4S-DTCWT of an image is formalised in the next few paragraphs.

Let I_t denote the base images for trees $t = 1 \ldots 4$ respectively. $I_1 = I$, the original image, and I is scaled by factors of 7/8, 6/8 and 5/8 to produce I_2, I_3, I_4 respectively, using bilinear interpolation. Let $H_{k_t}(x, y, d)$ be the complex wavelet coefficient at level k, location (x, y) and orientation d from the DTCWT of image I_t. The complex wavelet coefficients from the four different trees, denoted by $\widetilde{H}_{k_t}(x, y, d)$ are interleaved using the following relations

$$\begin{aligned}
\widetilde{H}_{4\,k-3}(x, y, d) &= \overline{H}_{k_1}(x, y, d) \quad \text{for } k_1 = 1 \ldots K \\
\widetilde{H}_{4\,k-2}(x, y, d) &= \overline{H}_{k_2}(x, y, d) \quad \text{for } k_2 = 1 \ldots K - 1 \\
\widetilde{H}_{4\,k-1}(x, y, d) &= \overline{H}_{k_3}(x, y, d) \quad \text{for } k_3 = 1 \ldots K - 1 \\
\widetilde{H}_{4\,k}(x, y, d) &= \overline{H}_{k_4}(x, y, d) \quad \text{for } k_4 = 1 \ldots K - 1.
\end{aligned} \tag{3.2}$$

The coefficients $\widetilde{H}_k(x, y, d)$ form the 4S-DTCWT scale-space. The sampling factors and interleaving are detailed in Table 3.1. 4S-DTCWT remains computationally efficient, requiring about $1 + (\frac{7}{8})^2 + (\frac{6}{8})^2 + (\frac{5}{8})^2 \approx 2.7$ times the computation required by the native DTCWT, plus the cost of the initial image resampling.

Figure 3.2 illustrates the extent to which 4S-DTCWT improves the scale invariance of the response function (3.4), by showing the responses arising from a set of 2D Gaussian shaped blobs of fixed amplitude and slowly changing spatial scale.

[1] This is not actually a minimally sampled wavelet transform, just a method for computing a scale-space pyramid.

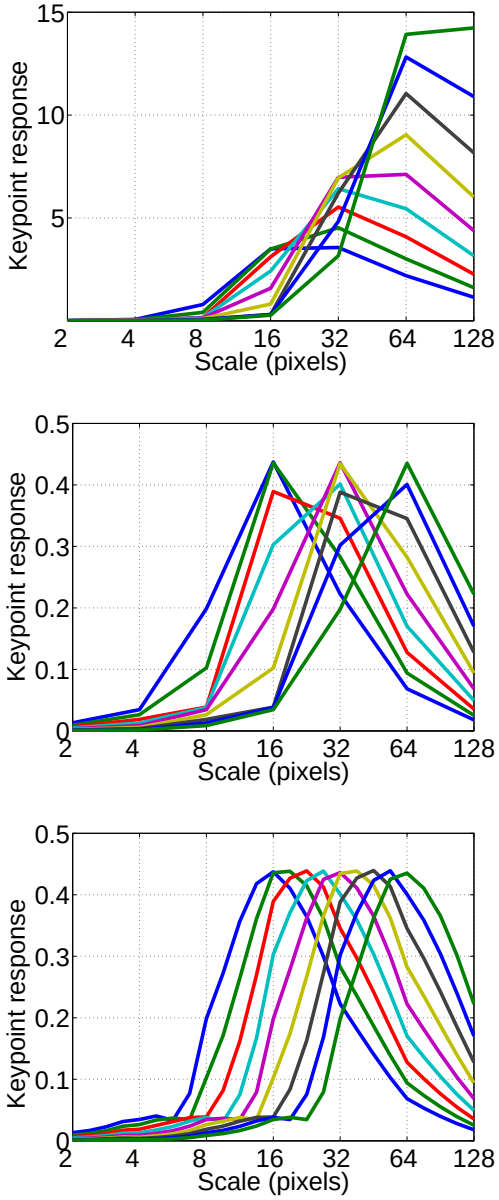


Figure 3.2: Scale responses to a set of images containing 2D Gaussian blobs with fixed amplitude and varying widths (σ=4 to 16 in steps of $2^{1/4}$). Each curve shows the response for a single image, plotted as a function of pyramid scale at the (x, y) peaks of the single scale response (3.4). *Top*: standard DTCWT without scale normalisation. *Middle*: standard DTCWT with 2^{-k} scale normalisation. *Bottom*: 4S-DTCWT, with both scale normalisation and denser scale sampling. Notice the extent to which 4S-DTCWT provides both better-defined peaks and more consistent response amplitudes over scale. This leads directly to more consistent scale estimates.

Resize factor ($1/s'_k$)	0.50	0.44	0.37	0.31	0.25	0.22	0.19	0.16
(as a fraction)	$\frac{1}{2}$	$\frac{7}{16}$	$\frac{6}{16}$	$\frac{5}{16}$	$\frac{1}{4}$	$\frac{7}{32}$	$\frac{6}{32}$	$\frac{5}{32}$
Level in Tree 1 (k_1)	1	.	.	.	2	.	.	.
Level in Tree 2 (k_2)		1	.	.	.	2	.	.
Level in Tree 3 (k_3)			1	.	.	.	2	.
Level in Tree 4 (k_4)				1	.	.	.	2
Level in 4S-DTCWT (k)	1	2	3	4	5	6	7	8
Scale (s'_k)	2.00	2.29	2.67	3.20	4.00	4.57	5.33	6.40
Log-Scale ($s_k = \log_2 s'_k$)	1.00	1.19	1.42	1.68	2.00	2.19	2.42	2.68

Table 3.1: The levels from four trees are interleaved to form the 4S-DTCWT. Each row in the central part of the table lists the levels for one of the four trees. The bottom part of the table lists the levels in the final '4S-DTCWT' tree, the corresponding scale (on linear as well as log-scale). Each column corresponds to a certain size (or resize factor) of the original image. For example, Level 1 of Tree 1 corresponds to half the size of the original image, Level 1 of Tree 2 corresponds to $7/16^{\text{th}}$ the size of the original image and so on. We do not have any responses at pixel resolution (resize factor = 1) because the DTCWT filters include a decimation by 2 operation. The shaded region shows the first octave in each tree. The symbol (.) means that there is no response corresponding to that resize factor. Note that each tree contains information at different scaling factors hence interleaving them gives us responses with denser sampling in scale.

3.2.3 How many trees are enough?

The effect of varying the number of levels per octave (the number of trees used in the decomposition) is illustrated in Figure 3.3.

While the advantage of using 4 trees over 2 trees or a single tree is evident (smoother responses and better scale localisation), there seems to be no real gain in using more than 4 trees. We also need to balance the number of features that we detect as a result of having more levels in a scale-space and the number of levels that we need to localise all of the resulting keypoints. In applications where speed is an important consideration and some loss in accuracy is acceptable, a 2-tree decomposition may be sufficient.

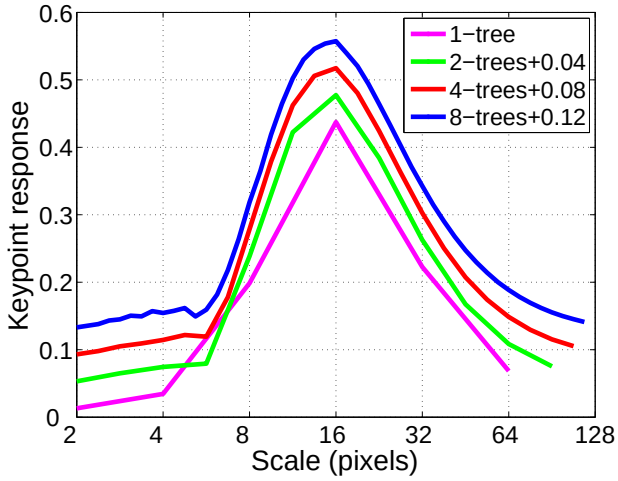


Figure 3.3: Scale responses for an image containing a single 2D Gaussian blob with fixed amplitude and width (σ) = 4. Each curve shows the response for a single image, plotted as a function of pyramid scale at the (x, y) peaks of the single scale response (3.4) The curves have been successively displaced by a small amount (0.04) along the y-axis for better visualisation. The curves from bottom to top show the response for 1-tree, 2-tree, 4-tree and 8-tree decomposition of the image. The 4-tree decomposition (red curve) corresponds to the 4S-DTCWT. While the advantage of using 4 trees over 2 trees or a single tree is evident, there seems to be no real gain in using more than 4 trees.

A similar study performed for SIFT using Gaussian smoothing to create the scale-space (pp 96-97 and Figure 3 in [Lowe, 2004]) concludes

> *"It might seem surprising that the repeatability does not continue to improve as more scales are sampled. The reason is that this results in many more local extrema being detected, but these extrema are on average less stable and therefore are less likely to be detected in the transformed image. The number of keypoints rises with increased sampling of scales and the total number of correct matches also rises. Since the success of object recognition often depends more on the quantity of correctly matched keypoints, as opposed to their percentage correct matching, for many applications it will be optimal to use a larger number of scale samples. However, the cost of computation also rises with this number, so*

> *for the experiments in this paper we have chosen to use just 3 scale samples per octave."*

Note that "3 scale samples per octave" in the above comment refers to the parameter 's' in SIFT. There are $s+2$ Gaussian images per octave and hence $(s+2)-1$ Difference of Gaussian images per octave. Therefore, $s=3$ for SIFT is analogous to 4S-DTCWT's 4-levels per octave. Experimentally, we find that 4-tree decomposition (4 levels per octave) gives the best tradeoff between the number of detections and computational efficiency.

3.2.4 Uniform sampling in scale

Ideally, we would like to sample uniformly over the range of scales available within an image. While this is desirable, the discrete nature of pixel grids makes such a sampling (or a very good approximation of such a sampling), a fairly involved task because

- The resize factors resulting from uniform sampling are often irrational numbers and lead to fractional image sizes in the scale-space pyramid.
- The DTCWT implementation requires the inputs to be a size that is a multiple of four.
- Any attempt to make the image sizes multiples of four involves zero-padding the image or chopping off the image close to the edges, resulting in artifacts.

Further, Figure 3.4 shows that the keypoint responses obtained with non-uniform sampling are very similar to those with uniform sampling. Therefore, the 4S-DTCWT serves as a good approximation for the 4-tree decomposition with uniform scale sampling. We avoid the need to rescale the DTCWT filters fractionally by choosing resizing factors that are fractions of 8, for example $[1, 7/8, 6/8, 5/8]$. While it is possible to overcome these problems and handle images of arbitrary sizes (with non-trivial programming effort), we make do with the solution presented here in order focus more on our main topic of research.

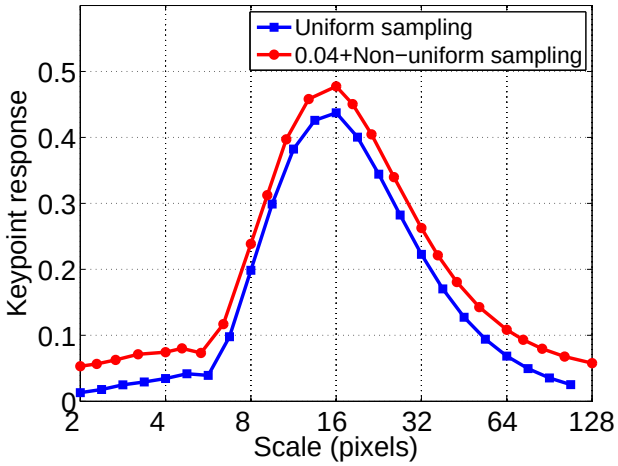


Figure 3.4: Scale responses for an image containing a single 2D Gaussian blob with fixed amplitude and width $(\sigma) = 4$. Each curve shows the response for a single image, plotted as a function of pyramid scale at the (x, y) peaks of the single scale response (The top curve is displaced by a small amount (0.04) along the y-axis for better visualisation). The red curve with circular markers corresponds to the 4S-DTCWT, *i.e.* non-uniform sampling in scale, while the blue curve with square markers corresponds to a 4-tree decomposition with a uniform sampling in scale. Within any octave, the samples on the blue curve are spaced evenly, while the distance between the consecutive samples on the red curve increases progressively. The shapes of the two curves are very similar, so the 4S-DTCWT serves as a good approximation to the 4-tree decomposition with uniform scale sampling.

SIFT gets around this problem by keeping all levels within an octave the same size as the base image for that octave. Thus the image sizing problems (or edge effects) have to be handled only once per octave in SIFT. In earlier versions of SIFT [Lowe, 1999], this problem was handled by choosing simpler factors for scaling, *e.g.* 1.5 instead of $\sqrt{2}$ along with bilinear interpolation.

While our strategy avoids duplication of data, we have to deal with a more severe constraint on image sizes, *i.e.* image size problems (or edge effects) have to be handled anew at each level, rather than at each octave (as in SIFT). We illustrate these differences in detail in Figure 3.5.

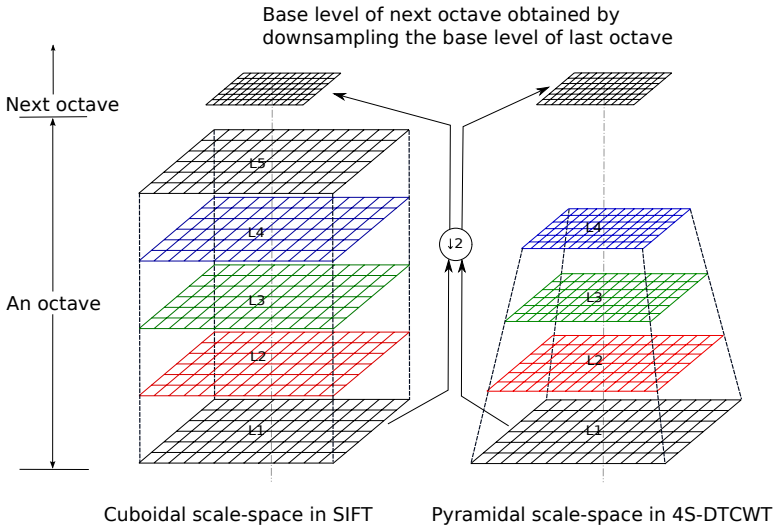


Figure 3.5: Cuboidal vs pyramidal scale-space. *Left*: The 'cuboidal' scale-space used in SIFT. The sizes of the levels are constant within an octave *Right*: The 'pyramidal' scale-space used in 4S-DTCWT. The sizes of the levels decrease successively within an octave. In both cases, the base image of the next (coarser) octave is the downsampled (by 2) version of the base image of the previous (finer) octave.

3.3 Corner strength measure

Although we tested a Harris-like single-level keypoint strength function of the geometric-mean (GM) form

$$\widetilde{E}_k(x,y) \;=\; \prod_{d=1}^{6} |\widetilde{H}_k(x,y,d)|^{1/6} \quad , \tag{3.3}$$

we prefer to use a minimum value (Min) function

$$\widetilde{E}_k(x,y) \;\equiv\; \min_{d\,\in\,\{1,\ldots,6\}} |\widetilde{H}_k(x,y,d)| \,. \tag{3.4}$$

The differences in the two cornerness measures are illustrated in Figure 3.6. The cornerness measure adopted (3.4) is somewhat analogous to

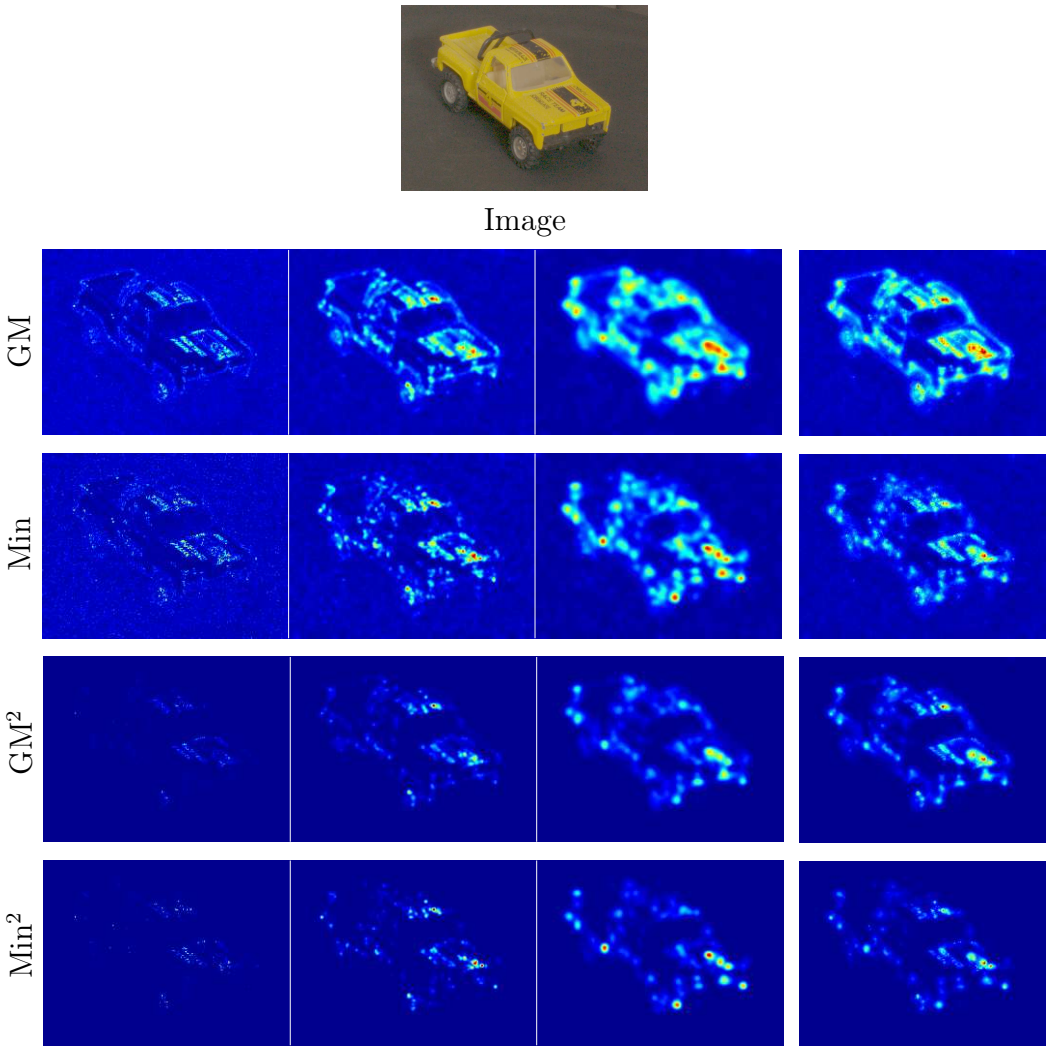


Figure 3.6: Different cornerness measures. Responses for the car image (top) for two cornerness measures: Geometric Mean (GM) in the top row and Minimum value (Min) in the second row. On left in each row, we have the responses at three levels that are one octave apart each (upsampled to be the same size as the finest of the three responses) and on the right, we have the result of adding these three responses. In the last two rows, we show the effect of squaring the responses. Notice the reduced edge responses in the accumulated maps when Min cornerness measure is used.

using the minimum eigenvalue of the oriented energy tensor[2] (Förstner de-

[2]The function $\widetilde{E}_k(x,y) = \prod_{d=1}^{6} |\widetilde{H}_k(x,y,d)|^{1/3}$ is analogous to $\widetilde{E}_k(x,y) =$

tector [Förstner, 1994]), as the discrete minimum over orientations spaced by 30° is similar to the continuous minimum over all orientations[3]. Conversely (3.3) is analogous to using the determinant of oriented energy (Harris detector [Harris and Stephens, 1988]). Overall the performance of the two approaches is similar but the one adopted gives slightly better rejection of false responses along edges, and is also faster to evaluate. Note that in both cases our detector is Harris-like in the sense that it finds regions (keypoints) that have strong oriented edge energy at multiple orientations, and thus sharply defined local autocorrelation functions. This is in contrast to methods (Hessian, SIFT, *etc.*) that key on "blob"-like structures but not on arbitrary strong 2D textures. Blobs occur less frequently than 2D textures, although their clearly defined positions and scales do make them particularly suitable for multiscale descriptor based matching. In practice both kinds of detectors are useful.

3.4 Keypoint localisation for a multi-scale detector

A good keypoint detection method should be capable of handling (at least) small dilations and shifts in the image. The detections should be distinct and reasonably well-localised in scale and space. To ensure this, some approach of combining information from different levels of the scale-space is necessary. Keypoints can be localised in the *accumulated map* or in *individual levels*. We discuss both the approaches and explain our choice of the latter.

$\prod_{d=1}^{6} |\widetilde{H}_k(x, y, d)|^{1/6}$ because one can be obtained from another as we know that the magnitudes of the complex wavelet coefficients are positive.

[3] We also attempted to estimate the angular minimum embodied in (3.4) more accurately using inter-orientation interpolation and using angular Fourier expansion, but this did not improve the resulting detector. It seems that 30° sampling is not fine enough to make interpolation over the necessary 3–4 adjacent samples a reliable predictor of the true response at intermediate orientations, but still fine enough to give good rejection of edge responses (*c.f.* Figure 3.6).

3.4.1 Localisation in accumulated map

The keypoint detection method in [Fauqueur et al., 2006] looks for points that are characterised by a large response in the accumulated map (accumap), A_1, a multiscale accumulation of the 'Geometric Mean' corner strength maps. This technique is used to combine information contained in corner strength maps from multiple levels of the DTCWT. During wavelet decomposition, coarser levels are created by decimating the finer level by a factor of 2. In order to combine the corner strength information from different levels, the coarser level accumap is upsampled by the same factor and then added to corner response at the finer level. This combination is then treated as the coarser level for the next step and the process continues until we reach the original image resolution. The rationale is that the dominant features in an image will be characterised by good persistence across levels, therefore adding the responses at different levels should enhance the stable features and suppress noise. The energy accumulation function $A_k(x, y)$ for any multilevel map is defined as,

$$\begin{aligned} A_{N-1}(x,y) &= \uparrow \widetilde{E}_N(x,y) \quad \text{and} \\ A_k(x,y) &= \uparrow \Big(\widetilde{E}_k(x,y) + A_{k+1}(x,y) \Big) \quad k = N-1, \ldots 1. \end{aligned} \tag{3.5}$$

where the $\uparrow$ operator represents upsampling by a factor of 2. Keypoints are detected at strong maxima of A_1. We make use of a strength threshold α, related to the height of the highest accumap peak, to decide which maxima are strong enough.

Limitations

It is difficult (as well as computationally demanding) to estimate the scale because the information about the level(s) at which corner strength was contributed is lost in the process of accumulation. This is illustrated in Figure 3.7. We investigate some approaches to scale estimation from accumulated maps in Section 3.5.1 and Section 3.5.2.

Consider a case in which two keypoints with very different scales are

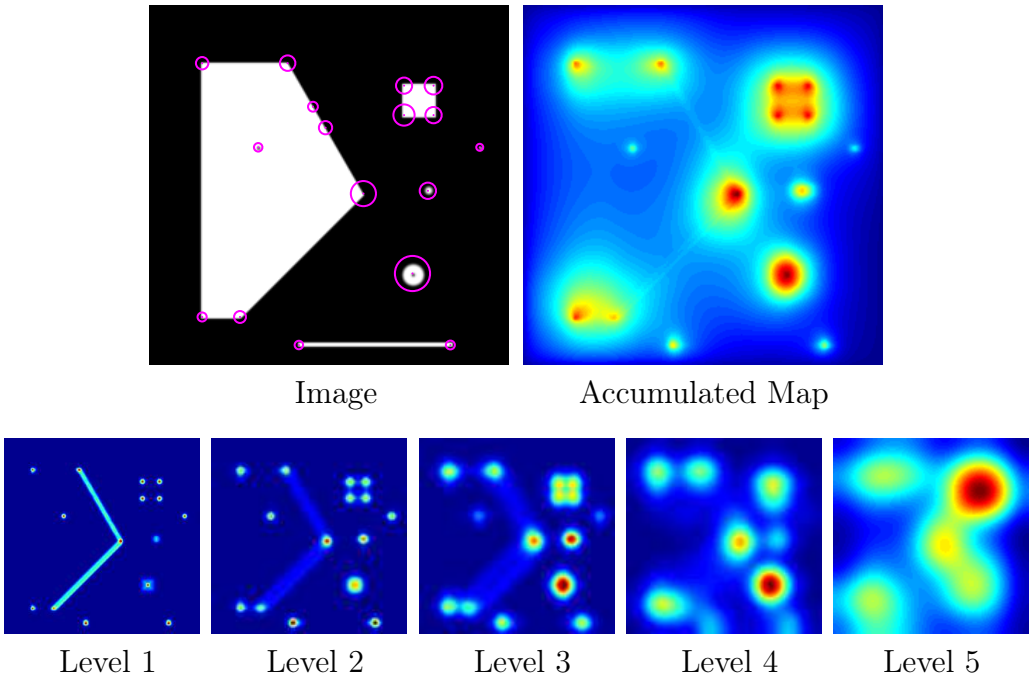


Figure 3.7: *Top*: A test image (left) and the corresponding accumulated map (right). The keypoints detected in the image are marked as magenta circles. *Bottom*: The Geometric Mean (GM) response at five consecutive octaves (finest level on the left). The responses have been multiplied by 2^{-k} (k = DTCWT level) for better visualisation. Warmer colours indicate larger values. For example, one would expect four fine scale keypoints corresponding to the four corners of the square and one coarse scale keypoint corresponding to the entire square to be detected. There is good support for the square to be identified as a coarse scale feature from levels 4-5, but the responses from levels 1-3 determine the positions of the peaks in the accumulated map. As a result, no coarse scale keypoint corresponding to the square is detected. The two concentric blobs should also be detected as two separate keypoints (one at a fine scale and one at a coarser scale) but only one keypoint is detected because they are co-located. It is clear from the keypoint responses at individual levels that each keypoint only exists over a *limited* set of scales.

located close to one another in the image. In the accumap approach, the contribution of the finer scale keypoint will tend to dominate the location of the peak, and hence the location assigned to both keypoints. However, for the coarser scale keypoint, only the information related to the coarse scale keypoint should be used to locate the coarse scale keypoint. Similarly, only

the information related to the finer scale keypoint should be used to locate the fine scale keypoint (*c.f.* Figure 3.7). Further, it is well known [Lindeberg, 1993, 1998] that the location of a corner-like keypoint varies in accordance with the scale or blur of the image. In an accumap, it is difficult both to detect keypoints with mid-range scale reliably and to determine their exact position and scale. Coarser scale features dominate the local height of the accumap (and hence the scale). Fine scale features dominate the local peaks of the accumap (and hence the spatial localisation).

3.4.2 Localisation in individual levels

We prefer to localise keypoints using individual levels, rather than accumulating responses across scales. This approach, (also used in SIFT [Lowe, 2004], HAR-AFF [Mikolajczyk et al., 2005], MOPS [Brown, 2005], DOLP [Crowley and Parker, 1984]) has the advantage that the keypoint's detection level changes in accordance with the scaling of the image content and it can be used as an initial scale estimate. In this approach, each level (single-scale response (3.4)) is searched in x-y space for a maximum to form a list of putative detections. Optionally, this may be refined to sub-pixel accuracy. Next, each putative detection is checked to ensure that it is also a clear local maximum in scale (interpolating the levels below and above the detection level where necessary). Finally, a subpixel interpolation mechanism is used to refine these to subpixel accuracy in position and sub-level accuracy in scale.

3.5 Scale estimation in keypoint responses at pixel resolution

In this section, we describe two scale estimation methods that operate on pixel resolution keypoint response maps. The first method, Steepest gradient, is often used in conjunction with the accumulated map approach to keypoint localisation (Section 3.4.1) and was used in FKA detector [Fauqueur et al., 2006]. The second method, Half-Max, will be used for keypoint localisation

in individual levels (Section 3.4.2).

3.5.1 Steepest gradient method

This section describes the method used in the FKA detector [Fauqueur et al., 2006] and we include it here for the sake of completeness. The scale of the keypoint is determined by the distance from its location at which the average negative radial gradient of the strength map has a strong maximum. This distance is found by projecting rays outward from the keypoint location in eight directions (multiples of 45°) and calculating gradients along these rays using forward differences. The distance at which the sum of the negative gradients over the eight directions reaches a maximum is marked as the scale of the keypoint.

This method was proposed for use with the accumap. There was an underlying assumption that there is no interference between the responses of different keypoints (*i.e.* the regions covered by different keypoints in the image are completely non-overlapping), which is rarely the case in real images. This problem is illustrated in Figure 3.7.

The implementation described in [Fauqueur et al., 2006] does not fully utilise the directional properties of the DTCWT as it samples the angular space more coarsely than the filters that make up the accumap. The scale estimates obtained from this method are not rotation-invariant because the directions in which the gradients are calculated are fixed. Denser angular sampling helps mitigate this problem. For example, we tried integrating accumap values along concentric rings centred at the keypoint to increase tolerance to rotational changes, but this is computationally demanding.

3.5.2 Half Maximum measure

Assuming that the keypoint responses are reasonably smooth and monotonically decreasing energy functions, the scale of a keypoint can be estimated by measuring the width of the response around the keypoint. We use the distance to the closest point at which the keypoint response falls to half of

its maximum value as the scale estimate. This makes the method relatively insensitive to the actual value of the response at the peak location.

The half maximum (half-max) distance is determined by performing a search starting at the keypoint location. We move outwards from the keypoint in successive steps of the search, but within each step, the order in which the pixels are searched is not an exact spiral. The pixels within each ring[4] are searched in a top-down then left-right fashion. Once the desired distance has been found, we use a final verification step using mean value over an approximately ring-shaped region to refine the estimate. This also makes the scale estimate slightly more robust to noise.

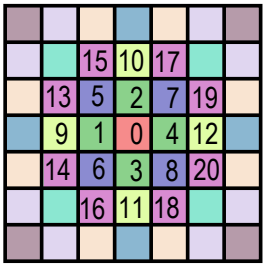


Figure 3.8: The search method used to search for the distance at which the response falls to half the peak value. The central pixel (keypoint) is the one marked '0'. Pixels at equal distances from the central pixel are marked by the same colour. The numbers denote the order in which pixels are tested. This arrangement is meant for arrays that use column-based indexing, and the arrangement is designed to minimise cache misses.

Since there is no preferential treatment to some directions over others, these scale estimates are more tolerant to rotational transformations. Instead of the half-max distance, one could also use the half-power distance. This strategy also gives fairly stable scale estimates. While the half-power scale measure is less affected by neighbouring image features than the half-max scale measure (by virtue of being smaller), it is affected more by the errors arising out of finite-sized square sampling grid. The choice of *half* power or *half* maximum point is rather arbitrary. As long as the scale estimates are such that they encompass the feature entirely, there is no methodical way

[4]The set of all pixels that are at the same distance from the keypoint is referred to as a 'ring' here.

of deciding which point one should choose as the scale estimate. However, this is a parameter that can be reliably optimised as shown in [Winder and Brown, 2007].

Although the keypoints are detected at each level separately, the scale estimation is still done on a pixel resolution response. The keypoint responses at all the levels have to be upsampled to pixel resolution. For large scale keypoints, the search for the half-max distance can be slow due to the large search regions, unless the search is explicitly constrained to the region corresponding to the scale of the level in which the keypoint has been detected. However, the size of the desired search region (at pixel resolution) increases quadratically with the level, making the search slow for coarse-scale keypoints. A search in a keypoint response at the detection level (no upsampling) will be faster, but this significantly impacts the accuracy of the scale estimation owing to reduced resolution of the data.

While in practice this method produces satisfactory results, we feel that it is sub-optimal because it takes into account the keypoint response information from only one scale at a time, necessitating some kind of non-maximum suppression step. In the absence of any such non-maximum suppression, this method can only be reliably used in conjunction with a detection method that uses information from multiple scales (*i.e.* the keypoints should be checked to ensure that they attain a clear maximum across scale). This is necessary in order to avoid multiple detections very close to one another at slightly different scales.

Non-maximum suppression:

If two keypoints are 'close enough', then only the stronger one (as per the keypoint response) is kept. Typically, a simple form of non-maximum suppression uses the Mahalanobis distance between the keypoints, viewed as points in the scale-space. The Mahalanobis distance is calculated as

$$||\mathbf{x}_i - \mathbf{x}_j||_{\mathbf{H}_i} = \left((\mathbf{x}_i - \mathbf{x}_j)^\top \mathbf{H}_i^{-1} (\mathbf{x}_i - \mathbf{x}_j)\right)^{1/2}. \tag{3.6}$$

Here, $\mathbf{H}_i$ is the diagonal covariance matrix such that

$$\text{diag}(\mathbf{H}_i) \;=\; \left[(s_i'\sigma_x)^2\,(s_i'\sigma_y)^2\,(\sigma_s)^2\right] \quad , \tag{3.7}$$

where $\sigma_{x,y,s}$ are the standard deviations of the peak in $x, y, \log_2(\text{scale})$ and s_i' is the scale in pixels of the i^{th} keypoint. This is similar to the use of $\mathbf{H}_i$ in the mean-shift process described in Section E.1. If the distance is less than a certain threshold, we keep the keypoint with the stronger keypoint response and discard the other one. Alternatively, the two could be replaced by their *mean* (mean location and scale). However, we do not do this because the mean keypoint is not guaranteed to be a maximum in scale-space. This process allows all keypoints that exist within a region of size $\sigma_x \times \sigma_y \times \sigma_s$ the chance to determine a single keypoint, thereby acting as a non-maximum suppressor.

3.6 Scale estimation in pyramidal scale-space

In this section, we describe two scale estimation methods that operate on pyramidal scale-space responses, Damped Newton and Local Least Squares. Both methods are used in conjunction with keypoint localisation from individual levels (Section 3.4.2). They operate directly on the pyramidal data, *i.e.* they do not require the keypoint responses to be upsampled to full pixel resolution.

After the initial keypoint localisation stage, we have a (possibly sub-pixel) keypoint location[5] and a scale initialisation that is obtained directly from the scale value associated with the level of the scale-space pyramid at which the keypoint has been localised. Keypoint responses are available only at grid locations at each level (quantised scale). Each grid point in the 4S-DTCWT scale-space has a quantised scale s, a spatial location (x, y), and a keypoint response associated with it. In both of the methods described below, given the discrete maximum (grid point in scale-space) and the data at all

[5]The sub-pixel keypoint location at this stage is a result of 2D spatial maximum interpolation only.

surrounding grid points, we are interested in estimating the sub-pixel position and the sub-level scale in scale-space that represents a stable maximum of the keypoint response in all three dimensions.

3.6.1 Damped Newton method

We use Damped Newton iterations to estimate the value and position of the maximum, given keypoint responses $f(\mathbf{x}_i)$ of a possibly-noisy keypoint response function $f(\mathbf{x})$, supplied at (possibly irregularly spaced) points $\mathbf{x}_i$. The keypoint responses from within a region $\pm 2\sigma_x \times \pm 2\sigma_y \times \pm 2\sigma_s$ centred on the detection are used to initialise the process.

Every grid point has a keypoint response $f(\mathbf{x}_i)$ associated with it and the i^{th} detection is denoted as $\mathbf{x}_i = [x_i \; y_i \; s_i]^\top$. We use expanding local coordinates centred at the discrete maximum (and at its subpixel positions at other scales). The scale coordinate is in $\log_2$(scale) and the coordinates expand in the sense that the grid spacing at all levels is $|\Delta x|, |\Delta y| = 1$.

Given a 3D Gaussian smoothing kernel $K(\mathbf{x}, \mathbf{x}_i)$ that has the form

$$K(\mathbf{x}, \mathbf{x}_i) \;=\; \exp\left[-\frac{1}{2}(\mathbf{x}-\mathbf{x}_i)^\top \Sigma^{-1} (\mathbf{x}-\mathbf{x}_i)\right] \tag{3.8}$$

in these coordinates, the subpixel estimate is produced by optimizing the normalized function $\hat{f}(\mathbf{x})$

$$\hat{f}(\mathbf{x}) \;=\; \frac{\sum_{i=1}^{n} f(\mathbf{x}_i)\, K(\mathbf{x}, \mathbf{x}_i)}{\sum_{i=1}^{n} K(\mathbf{x}, \mathbf{x}_i)}. \tag{3.9}$$

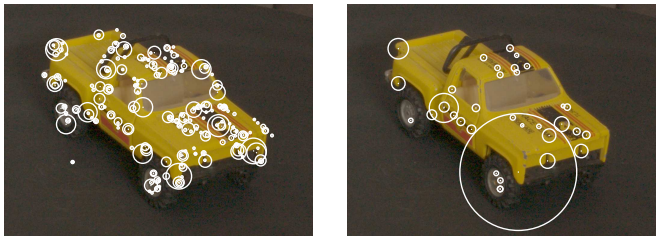
The normalization helps to compensate for the irregular sample spacing and it is here that the method improves on mean shift (see Appendix E.1) style ones[6]. We use a Damped Newton method to maximize $\hat{f}(\mathbf{x})$. At each iteration, we obtain a new estimate of the location of the maximum using

$$\bar{\mathbf{x}}(\tau+1) = \bar{\mathbf{x}}(\tau) - \hat{f}''^{-1}(\bar{\mathbf{x}}(\tau))\hat{f}'(\bar{\mathbf{x}}(\tau)) \tag{3.10}$$

[6]Mean shift tends to converge to more densely-sampled regions even if these have lower function values.

Spatial maxima from each 4S-DTCWT level that are also local maxima over scale, are used to initialise the process. A small diagonal damping ζ is added to the Hessian ($\hat{f}''^{-1}$) at each step to avoid problems with singularities. The iteration is continued to convergence $|\hat{f}(\bar{\mathbf{x}}(\tau+1)) - \hat{f}(\bar{\mathbf{x}}(\tau))| \leq \epsilon$, where ϵ is a small constant, typically around 10^{-6}.

Despite its elegance, we found that this method tends to require samples from a large number of levels (*c.f.* Figure 3.9), leading to high computational loads and frequent losses of detections owing to the peak locations drifting to much larger or smaller scales. As an alternative, we tested more local least squares quadratic fitting methods, as described in the next section.



246 keypoints 35 keypoints

Figure 3.9: Results of using the Damped Newton process with variable numbers of levels, with $\sigma_{x,y}$ set to 0.5 and σ_s set to 0.6. *Left*: Using three levels (the detection level and one above and below it) *Right*: Using thirteen levels (the detection level and six above and below it). We see that visually obvious features are merged or lost if the smoothing occurs over a wide range of scales.

3.6.2 Local Least Squares surface fitting

A common form of quadratic surface fitting is least squares fitting. Although 3×3×3 least squares fitting is often used for subpixel peak finding, its results do depend significantly on the sample chosen as centre (*e.g.* if several adjacent samples have essentially the same keypoint response) and on the kind of coordinate system used in the process. Here, we describe an adaptation of least squares fitting for subpixel peak finding in pyramidal data. Specifically, at each level of the pyramid we use the following steps:

- find 2D local maxima of the response function over 3×3 local patches,

- ensure that each of the 2D local maxima attains a clear maximum in scale with respect to the scale above and below
- extract a 3×3×3 block of scale-space function samples around the maximum by calculating the 3×3 patches at the location of the 2D maximum at the levels immediately above and below it,
- discard 2D maxima that are not local maxima over their full 3×3×3 patch,
- use least squares to fit a local quadratic function to the function samples on the patch, and
- use the peak of the quadratic function as the position, scale and value of the maximum.

The quadratic fitting is done in 'expanding' local coordinates around the 2D maximum, using log scale for the scale coordinate and (x, y) coordinates whose origin is the image location of the 2D maximum sample, transferred to its equivalent subpixel location at each of the three levels, and whose available (pyramid) samples are separated by $|\Delta x|, |\Delta y| = 1$ at each of the three levels. This rather strange local coordinate system is the best one to use for many local scale space computations. To the extent possible, it mediates between the fact that responses typically broaden in proportion to sample spacing as the scale changes, and the need to align corresponding image positions across different levels. Note that the samples at different levels are offset relative to one another and that adjacent levels are not uniformly spaced in scale in our pyramidal scale space (*c.f.* Figure 3.10). The least squares fit uses the exact position and scale values of each sample point in the expanding local coordinates to compensate for this.

Let $\mathbf{x} = [x, y, s]^\top$ be a point represented in the local coordinates centred on the keypoint. Here, (x, y) represent the spatial coordinates and s represents the $\log_2$ scale. We can define the quadratic function as

$$a + bx + cy + ds + ex^2 + fxy + gxs + hy^2 + iys + js^2 = q \tag{3.11}$$

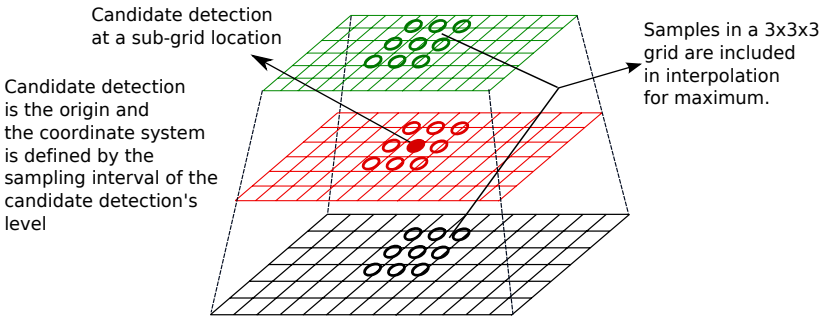


Figure 3.10: Local least squares fitting in expanding local coordinates. The candidate detection is the origin of the local coordinate system. The sampling interval of the candidate detection's level (middle red) is used as the reference spacing for the local coordinate system. Using this local coordinate system, the 3×3×3 grid is calculated and the keypoint responses at these grid points are also computed. A least squares surface fit is performed on the grid of data to determine the location of the maximum.

or, in matrix form as

$$\mathbf{d}\,\mathbf{p} \;=\; q \tag{3.12}$$

where $\mathbf{d} = [1\ x\ y\ s\ x^2\ xy\ xs\ y^2\ ys\ s^2]$ and $\mathbf{p} = [a\ b\ c\ d\ e\ f\ g\ h\ i\ j]^\top$ are the quadratic parameters.

In the 3×3×3 grid, we have 27 points. We can construct equations of the form of (3.6.2) for each of the 27 points and collate them. Let $\mathbf{D} = [\mathbf{d}_1; \ldots; \mathbf{d}_{27}]$ be the 27×10 matrix of $\mathbf{d}$ values at the 27 sample points and let $\mathbf{q} = [q_1; \ldots; q_{27}]$ be the corresponding vector of keypoint responses at these 27 points. The unequal spacing of the different scales is embedded in $\mathbf{D}$. The quadratic can be re-written as

$$\mathbf{D}\,\mathbf{p} \;=\; \mathbf{q} \tag{3.13}$$

and the best fit quadratic is given by the least squares solution to (3.6.2)

$$\mathbf{p} \;=\; \mathbf{D}^\dagger\mathbf{q} \tag{3.14}$$

where $\mathbf{D}^{\dagger}$ is the pseudo-inverse of $\mathbf{D}$. Once the quadratic parameters of the surface are known, we can estimate the peak location $\hat{\mathbf{x}}$ by equating the derivatives of the surface to zero and solving the resulting simultaneous equations,

$$\hat{\mathbf{x}} \;=\; -\mathbf{H}^{-1}\mathbf{g} \tag{3.15}$$

where

$$\mathbf{H} = \begin{bmatrix} 2e & f & g \\ f & 2h & i \\ g & i & 2j \end{bmatrix} \quad \text{and} \quad \mathbf{g} = \begin{pmatrix} b \\ c \\ d \end{pmatrix}. \tag{3.16}$$

The solution $\hat{\mathbf{x}} = [\hat{x},\ \hat{y},\ \hat{s}]^{\top}$ is the desired sub-pixel location and sub-level scale of the keypoint (expressed in expanding local coordinates with respect to the initial position of the maximum).

3.7 Issues in quadratic surface fitting

The problem that we are trying to solve, least squares function fitting, can be formulated as follows: Given points in (x, y, s), and values of the function $f(x, y, s)$ at these points, find the parameters of a function $f'(x, y, s)$ that best fits the function values $f(x, y, s)$ at the set of points (x, y, s). Figure 3.11 illustrates a problem with such a fit to determine sub-level scale. The quadratic does not have sufficient degrees of freedom to fit the data exactly and the fit is influenced more by the samples on the edges of the grid than by those at the centres. All points contribute equally but there are more edge points on each grid (eight compared to a single central point), so they have a greater influence on the fit. Owing to the relative breadths of the peaks at the different scales, this causes the fitted maximum to drift to coarser scales.

We present two ways to counter this effect. The first includes a spatial and scale-based weighting to give the centre pixel a bigger influence on the fit. The second ignores the edge points entirely doing a simpler 1D fit along the

scale dimension at the subpixel 2D position of the keypoint. This assumes that the maximum does not move much across scales and hence attempts only to refine the scale estimate. We now discuss both of these approaches.

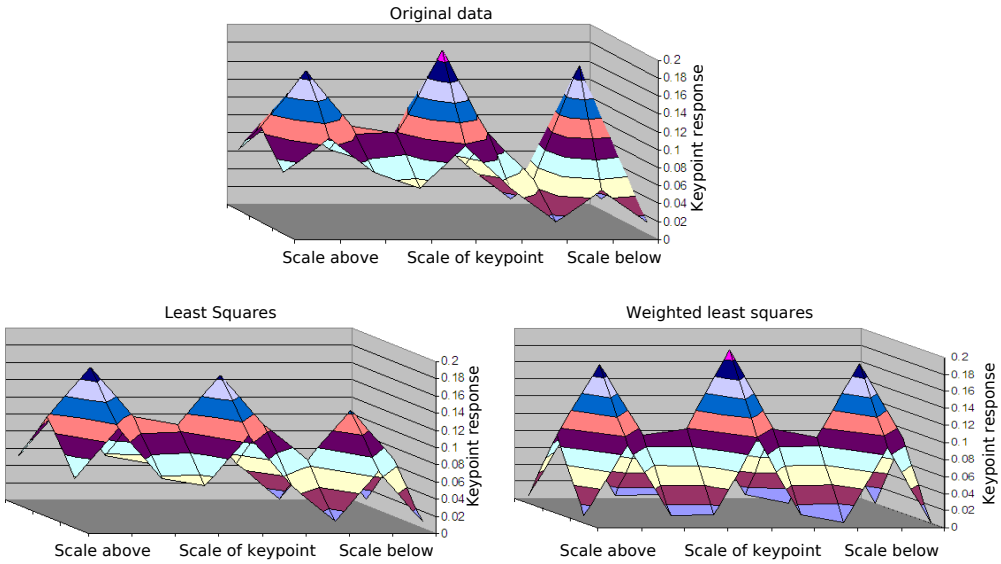


Figure 3.11: Surface plots of the keypoint responses at the three scales (denoted by the three peaks) for the original data (*top*), the quadratic fit (*bottom-left*) and the weighted quadratic fit (*bottom-right*). The quadratic fit is good at one level, but not at others. The weighted quadratic fit is good at points close to the centre of the 3×3×3 grid. The relative heights of the three centres of the spatial grids clearly show that the least squares fit does not sufficiently capture the nature of the original data.

3.7.1 Weighted least squares

First consider the use of weighted least squares function fitting to put more emphasis on the central point. If w_i are the weights of the 27 points on the 3×3×3 grid, then (3.6.2) may be modified as

$$\tilde{\mathbf{D}}\,\mathbf{p} \;=\; \tilde{\mathbf{q}} \tag{3.17}$$

where $\tilde{\mathbf{D}} = [w_1\mathbf{d}_1; \ldots; w_{27}\mathbf{d}_{27}]$ and $\tilde{\mathbf{q}} = [w_1 q_1; \ldots; w_{27} q_{27}]$. We used Gaussian weighting based on the distance of each point from the centre of the 3×3×3

grid. The result of using such a weighting is shown in Figure 3.13. It is also possible to use 2D Gaussian weighting with equal weights for all scales, but this is not found to be sufficient, an additional weighting for scales is needed.

3.7.2 Spline-fit

Taking our argument one step further, we ignore the neighbouring grid points completely, and instead fit a spline over the interpolated keypoint responses at the location of the central point at each level. We also take the opportunity to extend the number of levels to five. When projected to pixel resolution, these five points have exactly the same locations. Figure 3.12 shows the scale-space keypoint response for a Gaussian blob with $\sigma = 2^{2.25}$, a quadratic fit on keypoint responses over three levels, and a spline-fit over the entire curve. We see that the spline fit captures the rather asymmetric shape of the maximum significantly better than the quadratic fit. In practice, a spline-fit over five levels suffices. A cubic spline requires 4 points, and we want to have the same number of points on either side of the maximum, so we use 5 points. Due to the piecewise nature of spline fitting, using additional points would not improve the estimate of the maximum.

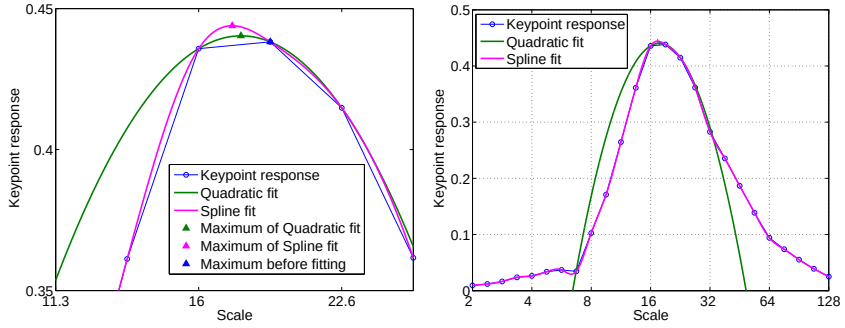


Figure 3.12: *Left*: A scale-space response for a Gaussian blob with $\sigma = 2^{2.25}$, a quadratic fit on keypoint responses over three levels and a spline fit. *Right*: A zoomed out version of the same with a spline fit over the entire curve. The spline-fit follows the shape of the scale response much more accurately than the quadratic fit, thus leading to more accurate scale estimates.

The above approach assumes that the spatial location of the maximum does not change with scale. This is true for blob-like features but less clear

for corner-like features. The idea of using the scale at which the scale response attains a maximum as the scale estimate for blobs was suggested in [Lindeberg, 1998] (see Section 5.1 and Figure 5) and has since been used in [Mikolajczyk, 2002, Mikolajczyk and Schmid, 2001, Mikolajczyk et al., 2005]. Our spline-fit method effectively does the same.

3.7.3 Discussion

Corner-like features may often shift in spatial position across scales. The spline fit method discussed above does not account for this. To our knowledge, apart from iterative spatial localisation and affine warping, no other solution for the simultaneous refinement of spatial position and scale in the full 3D space exists. A detailed discussion of this phenomenon can be found in Section 6.1, Section 7 and Figure 11 of [Lindeberg, 1998]. Lindeberg used a least squares formulation at a single scale to localise the corner. Furthermore, he studied the problems of choosing both the size of the area over which to perform least squares and a weighting function over this area. He also considered the problem of choosing the scale at which the single-level least squares is performed. His suggested solution was to use a Gaussian weighting that is proportional to the detection scale and perform a single-level least squares at multiple scales, then choose the scale at which the residual error of the fit normalised by the area covered at that scale is minimum.

A modification of the spline-fit described above would localise spatial maxima separately in the five scales and use these points to do the spline-fit instead of assuming that the spatial maxima at all five levels is at the same location. This may give a better overall fit, but it would have the additional computational overhead of having to search for the nearest spatial maximum to the keypoint in each scale and it might fail entirely if suitable maxima were not recovered.

Another option would be to use least squares with an elongated Gaussian weighting that can be rotated about the centre of the 3×3×3 grid and choose the weighting that best fits the data. This will allow movements in position as well as scale, but is slightly more involved than using a single orientation.

We have left these as future work.

3.8 Evaluation of scale estimation methods

In this section, we test the ability of the scale estimation methods discussed above to estimate scale correctly, across a wide range of scaling of the image content. To reduce spurious effects due to boundaries and neighbouring features, we base our tests on an idealized case, a single 2D Gaussian blob of fixed amplitude and variable scale located at the centre of the image. The image is 1024×1024 and the width (standard deviation σ) of the Gaussian is varied from 4 to 16 pixels in steps of $2^{1/32}$. We expect the estimated scale to vary linearly under changes in σ. Scale and σ are measured in the $\log_2$ domain.

Figure 3.13 shows the results. The thin pink line at 4σ is an empirically estimated baseline. In plot (a), we show the result of using the detection level as the scale estimate without any further refinement. This represents the *worst-case* scale estimate. Any valid refinement method must do better than this. In plot (b), we show the half maximum method, which is not better than using the detection level. Of particular concern is the decreasing scale estimate in response to an increasing σ. In plot (d), we have the result for 11-level Damped Newton process, which shows an improvement over the detection level, but the abrupt jumps between levels are still a cause of concern. Plot (e) shows the result of local least squares without any weighting. This is better than both the half-max and the Damped Newton process, but there are still some drifts at the coarsest end within each interval of levels. Note also that there are finer steps towards the finer end than the coarser end within each interval of levels. Plot (f) shows the result of local least squares with Gaussian weighting and plot (c) shows the result of doing a spline fit on the keypoint responses at the central point from 5 levels. This is better than our baseline result in plot (a). The estimated scale varies roughly linearly with respect to changes in σ. All of the scale estimates are refined to their actual value, thus overcoming the irregularities of non-uniform sampling in scale. The spline-fit method, (shown in plot (c)) and the least squares meth-

ods significantly improve the scale estimates and overcomes the difficulty posed by the non-uniform sampling in 4S-DTCWT scale-space.

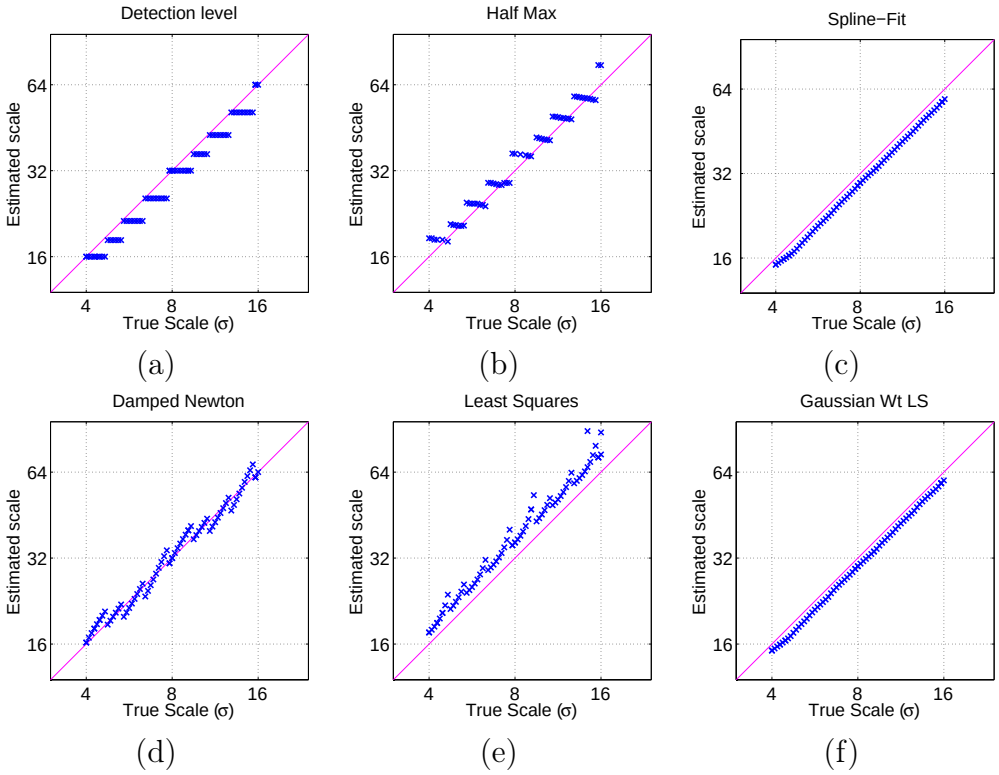


Figure 3.13: True scale vs estimated scale for a variety of scale estimation methods. The input in each case is a set of images containing 2D Gaussian blobs with fixed amplitude and varying widths (σ=4 to 16 in steps of $2^{1/32}$). Each point in the graphs shows the estimated scale for an image with a single Gaussian blob against the standard deviation σ of the input Gaussian blob. Scale and σ are measured in the $\log_2$ domain. We expect the estimated scale to vary linearly with respect to change in σ.

3.9 Qualitative evaluation

In this section, we present exemplar results for the scale estimation methods presented in Sections 3.5-3.6. The FKA detector uses Steepest gradient method for scale estimation. The keypoints localised in individual levels in the 4S-DTCWT scale-space use the Half-Max, Damped Newton, Local Least

Squares or the Spline fit method for scale estimation. In order to perform a qualitative evaluation of the above-mentioned methods, we show examples of keypoint detection on a single image for two settings of the minimum strength threshold α in Figures 3.14-3.18. A valid keypoint must have keypoint response greater than $\alpha\times$(maximum keypoint response at the level) in order to be kept.

From Figures 3.14-3.18, we conclude that the Weighted Least Squares method lead to a good compromise between stable scale estimates and speed of computation. This choice is illustrated further in Section 5.4.1. With this conclusion in mind, we present a summary of our preferred keypoint detector in the next section.

We have also tested the repeatability (with respect to change in viewpoint) for all these methods on a subset of our new 3D dataset. The experimental setup is described in detail in Chapter 5, hence we defer the discussion of the quantitative results until Section 5.4.1 in Chapter 5.

3.10 Overview of final BTK keypoint detector

The proposed keypoint detector (BTK) consists of

4S-DTCWT scale-space pyramid for denser scale sampling and reduced scale-related aliasing errors;

Min cornerness measure for better corner localisation, reduced edge response and speed of computation;

Maximum across scale for rejection of spurious spatial maxima;

Keypoint localisation in individual levels for better spatial localisation and reliable initialisation of the scale estimate; and finally

Weighted least squares for keypoint location and scale refinement in expanding local coordinates.

One may use the Spline fit method instead of weighted least squares for scale refinement and obtain similar results. Spline fit method is simple but does not include any position refinement.

In this chapter, we have developed a scale-space framework and a multi-scale keypoint detector based on the DTCWT. We have investigated several scale-space localisation methods. In Chapter 5, we will show that, as a result of this work, the BTK keypoint detector demonstrates position and scale stability comparable to most other popular detectors. For now, we take a minor detour and study some aspects of keypoint description and matching in Chapter 4.

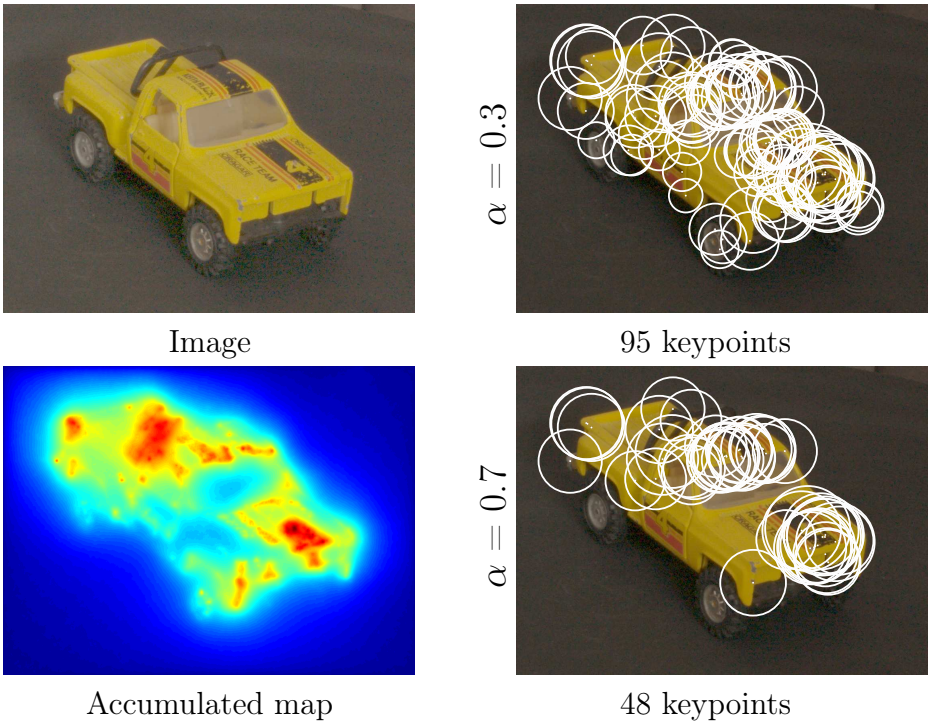


Image | 95 keypoints

Accumulated map | 48 keypoints

Figure 3.14: The results of scale estimation using steepest gradient method. *Left*: An image and the corresponding accumulated map. *Right*: The detected keypoints. We see that there is considerable interference between responses from nearby features, making it difficult to isolate the extent and hence the scale of any given feature. See, for example, the top surface of the car and the bonnet of the car with the digit '4' on it. Well-isolated features such as the front-left corner (above the front-left wheel) or the corner on the bottom-rear end are not affected by any nearby features and tend to be stable. An increase in the strength threshold should allow stronger keypoints to be selected. Instead, owing to the non-localised responses, the strength threshold tends to select only features in areas of high activity and leave out isolated features (*e.g.* the corners next to the digit '4' on the left side door of the car). Another point to be noted is the limited range of scales detected in the image.

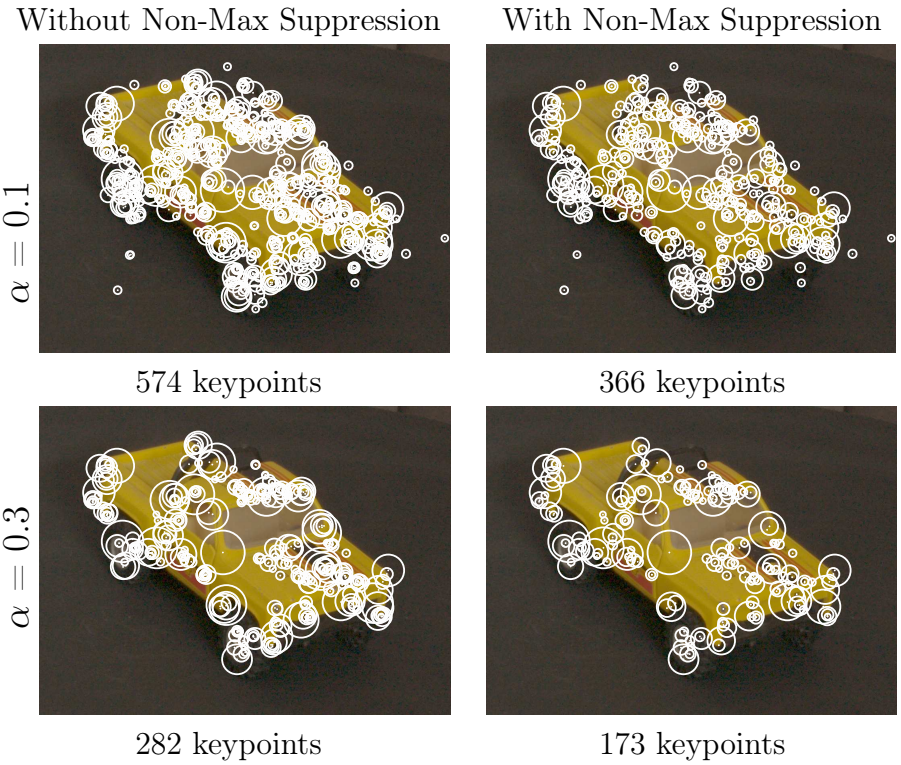


Figure 3.15: The results of scale estimation using Half Maximum method. *Left*: Without non-maximum suppression. *Right*: With non-maximum suppression. As expected, increasing the strength threshold eliminates keypoints on edges and regions of low activity (turn-table). Also, the use of non-maximum suppression reduces the number of multiple detections of the same feature at slightly different locations and scales, with a small computational overhead. This method is computationally expensive relative to other methods tested that give comparable results. Nevertheless, it benefits from the BTK detector's denser sampling in scale and localisation of keypoints at individual levels.

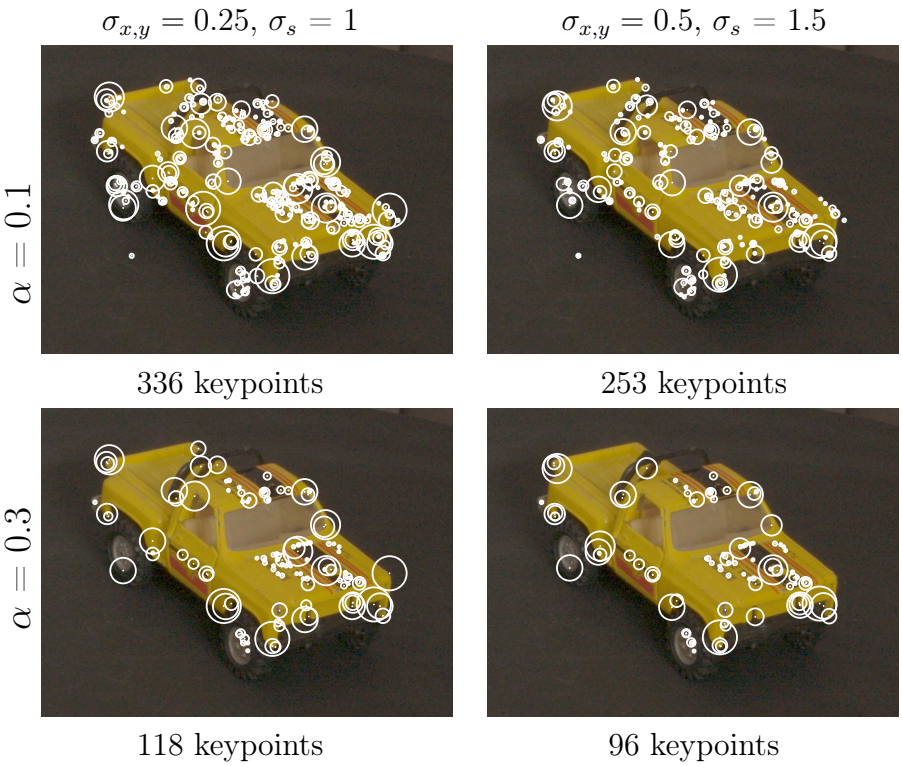


Figure 3.16: The results of scale estimation using Damped Newton process for two different sizes of the Gaussian smoothing kernel. *Left*: $\sigma_{x,y} = 0.25\times$Scale, σ_s=1 octave. *Right*: $\sigma_{x,y} = 0.5\times$Scale, σ_s=1.5 octaves. All keypoint responses within a region $2.5\times\sigma$ contribute to the Damped Newton process. We see that the keypoints retained after the Damped Newton process are much better localised in space and scale as compared to the steepest gradient method and the Half-Max method, *i.e.* there are fewer instances of the detector firing multiple times for a single feature. Furthermore, it behaves well under changes of strength threshold and generates a good spread of detection scales. Finally, we note that a larger smoothing kernel results in better separated keypoints in scale and space. The Damped Newton process has a computational complexity that is linear in the number of initial detections. It is faster than the Half-Max method for similar numbers of keypoints.

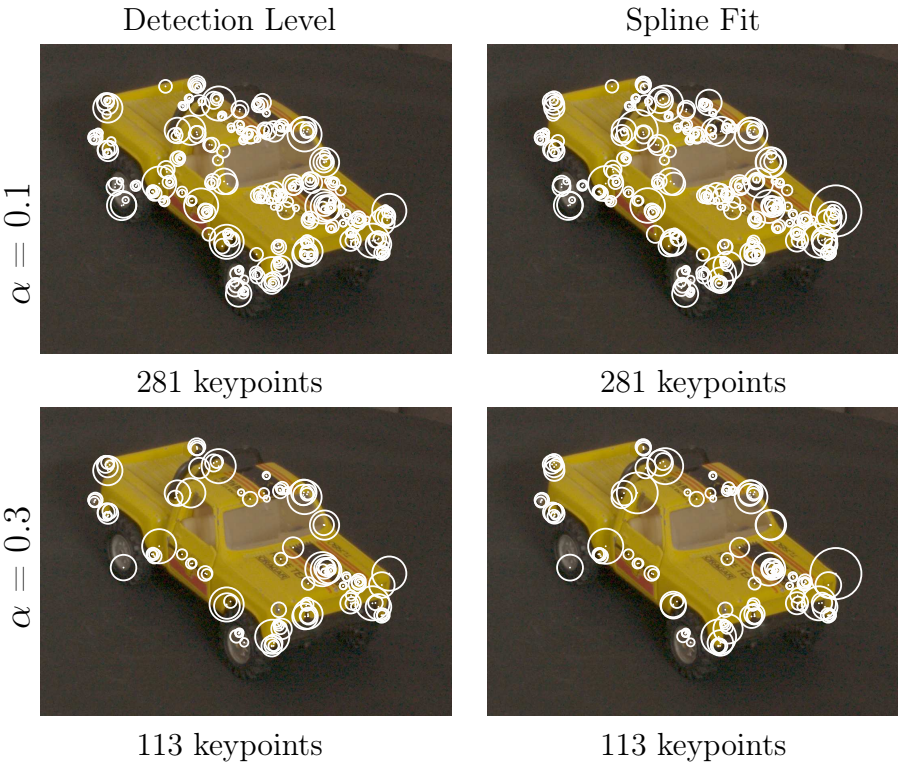


Figure 3.17: The results of scale estimation using the Spline fit method. *Left*: Detection Level is used as scale *Right*: Detection level is refined using spline fitting. We see that the keypoints found by the spline fitting method improve the scale estimate over the detection level. For example, note the keypoints on the bottom right corner of the front screen, the keypoints are found at multiple scale initially, but refined to the same scale after spline fitting. Also note that at the rear left corner of the car, several keypoints for the corner-like feature converge to similar scales.

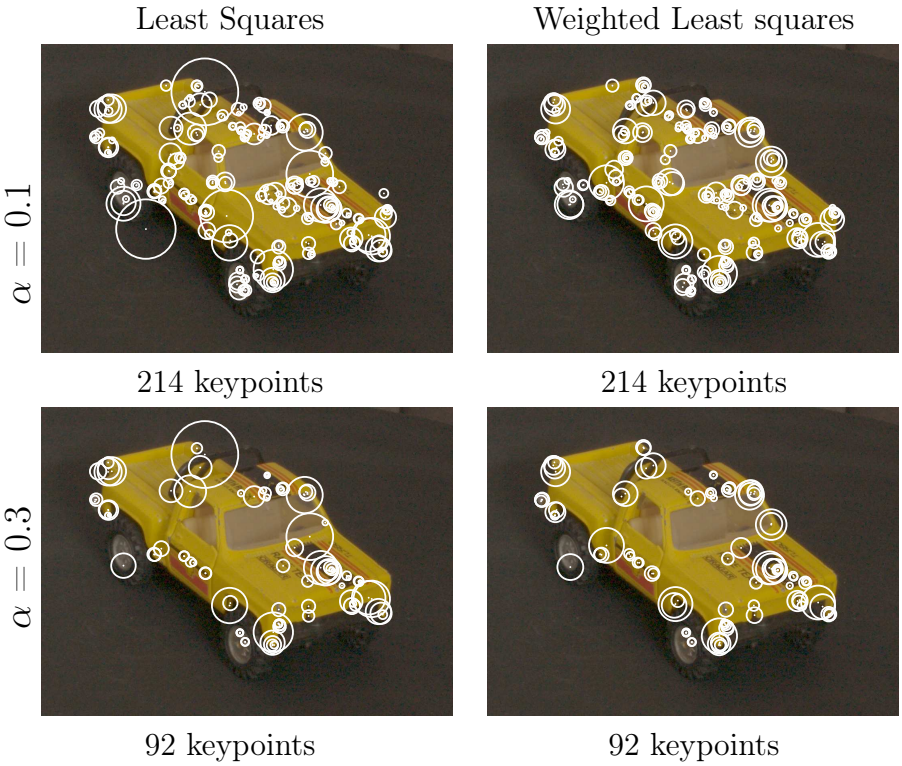


Figure 3.18: The results of scale estimation using the local least squares and weighted least squares methods. *Left*: Least squares *Right*: Weighted least squares. We see that the keypoints found by the local least squares quadratic fit are well localised in space and scale. The least squares method gives similar results to spline fit method, but has some position refinement capability embedded in it, hence is preferable.

Chapter 4

Keypoint descriptor and matching

To complement our detector we use the Polar matching matrix, a rotation-invariant local visual descriptor based on the DTCWT [Kingsbury, 2006]. We briefly present this descriptor, then adapt it for use with the BTK keypoints.

4.1 12×8 P-matrix descriptor

Polar matching matrix (**P**-matrix) descriptors are created from DTCWT coefficients as follows [Kingsbury, 2006]. At a designated DTCWT level and sampling radius, a circle of 12 points spaced 30° apart is placed around the central point (keypoint). For each DTCWT orientation, the complex DTCWT coefficients are evaluated at these points using spatial interpolation[1]. At each point 6 independent orientations at intervals of 30° (and their complex conjugate pairs along diametrically opposite directions) are available. The resulting coefficients are arranged in a 12×6 complex matrix. Within this matrix, column c contains the coefficients whose orientation, relative to the tangent to the sampling circle, at the sample position is $(30\,c - 15)°$.

Rotations of the image by multiples of 30° thus produce cyclic shifts within each column of the matrix, *i.e.* simple phase changes of the FFT of the column. This property allows efficient rotation-invariant descriptor com-

[1]Such interpolation is reliable owing to the band-limited nature of the rotationally symmetric DTCWT.

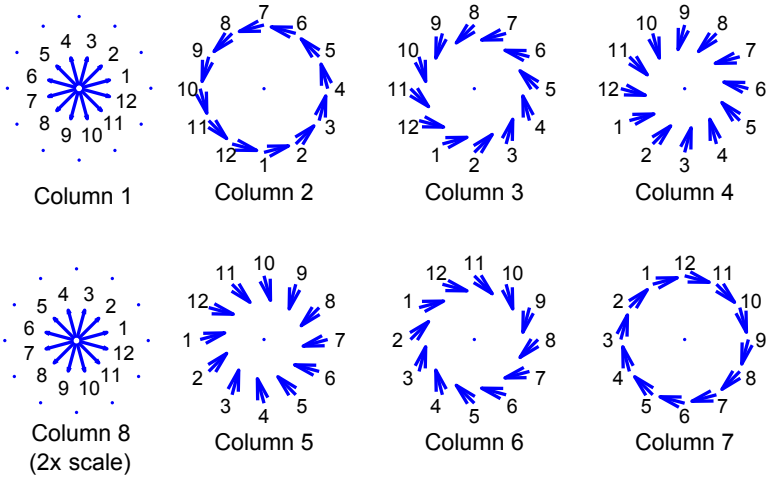


Figure 4.1: The construction of our 12×8 **P**-matrix descriptor. The arrangement is the same as in [Kingsbury, 2006], but we base the descriptor on the 4S-DTCWT. The small numbers (and the orientations of the arrows) denote the subband selected at each sampling location. Each column is composed of a set of rotationally symmetric samples. The figure is taken from [Kingsbury, 2006].

parison and efficient estimation of the relative rotations between matching descriptors. To produce a complete **P**-matrix descriptor, matrices from several circles with different radii and/or wavelet scales (tree levels) can be appended, and additional columns can be included based on the coefficients of the 12 orientations (6 conjugate pairs) at the central point at a given level. One of the most common arrangements [Kingsbury, 2006] is a spatially-compact local descriptor whose 12×8 matrix contains the coefficients from the central point at the scale of the keypoint in the first column, the coefficients from the ring at the scale of the keypoint in the next 6 columns (which is composed of the the 12×6 arrangement described above), and the central point at the next level up (wavelets 2× coarser) in the eighth column – see Figure 4.1. For illumination invariance, the total energy in each **P**-matrix is normalised to one, so matching them produces a correlation score in the range $[-1, 1]$.

4.2 Support for finer scale sampling

To use this descriptor with our detector, we need to adapt it to subpixel position and scale estimates. We use the 4S-DTCWT for the descriptors as well as the detector so raw coefficients are available at steps of around $2^{1/4}$ in scale and at integer locations at these scales. Given a keypoint, we lay out a circle of subpixel sample points corresponding to its exact subpixel location and scale (the circle having unit radius at this scale). We then build the 12×8 **P**-matrix descriptor by taking wavelet coefficients from the discrete 4S-DTCWT level whose scale is closest to the keypoint scale and using subpixel interpolation to estimate the wavelet responses at the designated sample points. For the 8^{th} column, we use the level 2× coarser. The descriptor is thus evaluated using wavelet coefficients from the nearest discrete scale, but at sample points corresponding to its exact subpixel position and (continuous) scale. We will refer to the 12×8 4S-DTCWT **P**-matrix descriptor as 'BTK descriptor'.

The BTK descriptor has some similarities to other modern descriptors such as SIFT and DAISY [Winder and Brown, 2007]. It is based on oriented energies at a set of spatial positions, it incorporates multiple scales and good illumination invariance, and it has an effective (and original) mechanism for handling the orientation degree of freedom. Although (with the current detector) it is not affine-invariant, like SIFT it tolerates small errors in keypoint positions and scales and small affine deformations relatively well. Figure 4.2 illustrates that under increasing affine deformations, BTK descriptor matching scores show degradations similar to those for SIFT, but perhaps slightly less rapid degradation at small deformations.

4.3 12×15 P-matrix descriptor

Figure 4.3 shows a 12×15 arrangement giving an alternative BTK descriptor. This consists of two rings at radii r and $2r$ at the keypoint scale, the central point at the keypoint scale and at scales 2× coarser and 2× finer. This allows slightly more tolerance to shifts away from the keypoint while encoding some

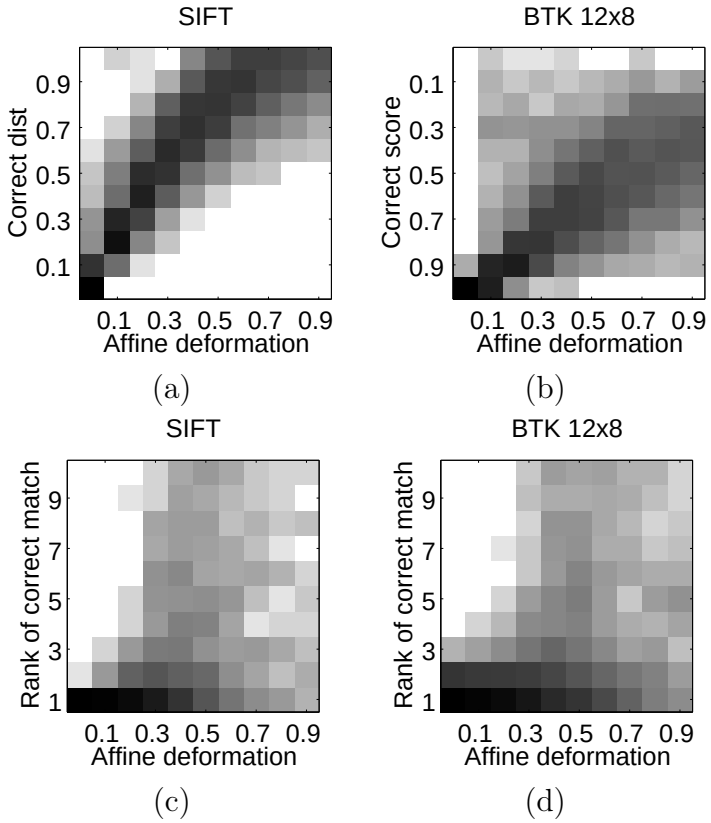


Figure 4.2: Descriptor mismatch under affine deformations. SIFT and BTK descriptors are computed and matched over identical (Difference-of-Gaussian) keypoints for increasing affine distortion (vertical shear) of a Graffiti image from the Oxford dataset [Mikolajczyk et al., 2005]. *Top row*: The two plots (a-b) show histograms of the resulting SIFT distances (0 is best) and DTCWT correlation matching scores (1 is best). *Bottom row*: The two plots (c-d) show histograms of the rank of the correct match using these respective distance metrics for ranking. In all cases, darker colours indicate higher bin counts with darkness proportional to log(count). Note that the distribution of ranks for BTK is similar to that for SIFT.

additional spatial information into the descriptor. The 12×15 descriptor exhibits better tolerance to shifts in the outer ring as compared to the inner ring, while allowing a richer description at a small computational overhead.

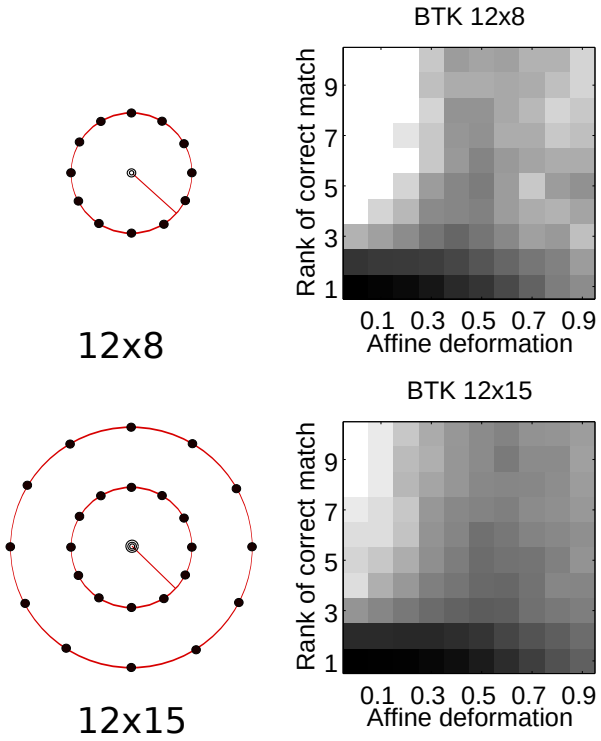


Figure 4.3: Comparison of different configurations of the BTK descriptor. The dots represent the sampling points and the rings represent the size of the sampling circle at pixel level. The ring with a radial line has a radius equal to the scale of the keypoint. The experimental setup is the same as in Figure 4.2. Figure 4.2-(d) is the same as Figure 4.3-(a).

4.4 Fast descriptor computation

We wish to calculate descriptors centred at subpixel locations using a circular pattern, which we call the descriptor grid. The available DTCWT responses are on a rectangular grid, called the response grid. We estimate the complex-valued DTCWT coefficients on the descriptor grid by interpolating values from the response grid separately in each subband. We use the method of Kingsbury [Kingsbury, 2006]. The interpolation must be band-limited and localised within the support region in 2D frequency space of the subband being interpolated. If the centre frequency of support region of the subband being interpolated is $\{\omega_1, \omega_2\}$, the band-limited interpolation may be performed as follows (*c.f.* [Kingsbury, 2006]):

FOR EACH KEYPOINT, DO

FOR EACH SUBBAND, DO

1. A frequency shift by $-\{\omega_1, \omega_2\}$, *i.e.* complex multiplication by $e^{-j(\omega_1 x_1 + \omega_2 x_2)}$, at each point (x_1, x_2) on the response grid.
2. Bi-cubic interpolation to estimate responses at the descriptor grid points (y_1, y_2) given the responses at response grid points (x_1, x_2).
3. An inverse frequency shift by ω_1, ω_2, *i.e.* complex multiplication by $e^{j(\omega_1 y_1 + \omega_2 y_2)}$, at each point (y_1, y_2) on the descriptor grid.

END

END

Note that Step 1 involves a $W/2^k \times H/2^k$ complex multiplication[2] while the complex multiplication in Step 3 involves 13 points for a 12×8 descriptor. Each subband at any level of the DTCWT can be interpolated independently of all other subbands and levels. The number of keypoints in an image is usually much greater than the number of levels in the pyramid so there are many keypoints per level. Obviously, they share the first step in the bandpass interpolation process described above. Further, this step is independent of the exact location of the descriptor grids on the response grid (*i.e.* it is independent of (y_1, y_2)). Therefore, it is more efficient to concatenate all the descriptor grids into one long list, so that Step 1 is performed only once per subband at each level.

These fairly straight-forward modifications produce a very significant speedup in the descriptor computation (*c.f.* Table 4.1). The timings are taken on the same machine (2.4 GHz Intel CPU with 2GB RAM) and averaged over multiple runs of each method. The measurement is always between 1-10 seconds for fast method and between 1-10 minutes for the old method.

[2]Typically, the keypoints occupy a sufficient proportion of the image, that isolating just the points that are needed for interpolation on the response grid is not worthwhile.

The measurement is divided by the number of runs and the number of total keypoints to obtain the time for each descriptor. Both the methods are implemented in MATLAB.

Total keypoints	10^3	10^4	10^5
Old method	5.7 ms/kp	5.69 ms/kp	5.67 ms/kp
Fast method (1 level)	2.1 μs/kp	1.05 μs/kp	0.97 μs/kp
Fast method (10 levels)	15 μs/kp	2.08 μs/kp	1.06 μs/kp
Fast method (35 levels)	39 μs/kp	5.63 μs/kp	1.55 μs/kp
Fast method (100 levels)	110 μs/kp	11.49 μs/kp	2.08 μs/kp

Table 4.1: Time required to calculate descriptors for a total of 10^3, 10^4 or 10^5 keypoints. The time taken by the fast method depends on the number of different levels at which the keypoints exist and on the total number of keypoints. Each entry in the table lists the time taken to compute a single descriptor. For example, if there are 100 keypoints each at 10 different levels (a total of 1000 keypoints), then each descriptor is computed in 15 μs by the fast method.

Realistically, even if one has a 10 mega-pixel image, the maximum number of octaves covered in the scale-space of such an image would be about nine[3]. In the 4S-DTCWT framework, this leads to at most 36 levels, so we have highlighted the most realistic scenario in Table 4.1, which lists timings for 35 levels. The case where there are 100 different levels is unrealistic at present, but included here for reference.

4.5 Fast descriptor matching

In this section, we describe the modifications we made to the FFT based algorithm proposed in [Kingsbury, 2006] for keypoint matching. First, we introduce some notation. We then describe the FFT based method (Pairwise method), followed by a description of our proposed method (Anglewise method) and finally, a comparison of the two methods.

[3]If maximum possible image dimension is less than $2^{12} = 4096$ and the coarsest level is at least $2^3 = 8$ wide, then we can go down about 9 octaves

Notation

For any matrix $\mathbf{A}$, we denote it's element-wise conjugate by $\mathbf{A}^*$, it's real part by $\Re(\mathbf{A})$ and it's vectorised form by $\text{vec}(\mathbf{A})$. It follows that the transpose of $\mathbf{A}^*$ can be written as $\mathbf{A}^{*\top}$. A column-wise Fast Fourier Transform (FFT) operation on $\mathbf{A}$ is written as $\text{FFT}(\mathbf{A})$ and the Inverse Fast Fourier Transform operation on $\mathbf{A}$ is written as $\text{IFFT}(\mathbf{A})$. The expression $\mathbf{s}\cdot\mathbf{r}$ denotes the element-wise multiplication of the vectors $\mathbf{s}$ and $\mathbf{r}$.

4.5.1 Pairwise method

The pairwise method for FFT based cross-correlation was proposed in [Kingsbury, 2006]. Given $\mathbf{P}_1$ and $\mathbf{P}_2$, the column-wise FFT of two $\mathbf{P}$-matrices, the correlation score $\mathbf{C}(\theta)$ is calculated as

$$\mathbf{C}(\theta) = \Re \text{ IFFT} \left\{ \sum_{\text{rows}} \mathbf{P}_1 \cdot \mathbf{P}_2^* \right\} \tag{4.1}$$

Here $\mathbf{P}_2^*$ is the element-wise conjugate of $\mathbf{P}_2$. For a finer angular resolution, one may do a four-fold extension of the element-wise product, $\mathbf{P}_1 \cdot \mathbf{P}_2^*$ by zero-padding the additional (high frequency) FFT elements, before doing the IFFT. The cost of one 48-point cross-correlation is $\approx \mathcal{O}(48 \log_2 48) + K$ complex multiply-adds where K is the descriptor size. A cross-correlation operation on each pair of $\mathbf{P}$-matrices provides us with matching scores for a pair of keypoints for 48 relative orientations. The angular resolution of such correlation scores is 7.5°.

4.5.2 Anglewise method

Here we describe an alternative way of computing the correlation scores in an environment which has been optimised for matrix computations, such as MATLAB, which achieves a significant speedup.

Let the number of relative orientations at which we wish to compute correlation scores be D so that the angular resolution is $\frac{360}{D}^\circ$. Typically D is 12 or 48. Let $\mathbf{s} = \text{vec}(\mathbf{P}_2)$ and $\mathbf{r} = \text{vec}(\mathbf{P}_1)$. Both $\mathbf{r}$ and $\mathbf{s}$ are column

vectors of length K (96 for a 12×8 **P**-matrix). Let $\mathbf{W} = [\mathbf{w}_1^\top\ \mathbf{w}_2^\top \ldots \mathbf{w}_D^\top]^\top$ be the $D{\times}K$ IFFT operator[4] ($\mathbf{w}_i$ is the i^{th} row of $\mathbf{W}$). We can write the cross-correlation operation in the following form, which allows us to compute the correlation score at each orientation independent of other orientations,

$$\mathbf{C}(\theta_i) \quad = \quad \Re\left\{\mathbf{w}_i^\top \cdot \mathbf{s}^* \cdot \mathbf{r}\right\} \tag{4.2}$$

Next, by making a substitution $\widehat{\mathbf{s}}_i^* = \mathbf{w}_i^\top \cdot \mathbf{s}^*$, we can express the above as a simple matrix multiplication of $\mathcal{O}(1{\times}K{\times}1)$

$$\mathbf{C}(\theta_i) \quad = \quad \Re\left\{\widehat{\mathbf{s}}_i^{*\top}\mathbf{r}\right\} \tag{4.3}$$

Further, if the weighted descriptor vectors $\mathbf{s}_i^*$ for many keypoints in one image are concatenated into a matrix $\widehat{\mathbf{S}}_i^* = [\mathbf{w}_i^\top \cdot \mathbf{s}_1^*\ \ \mathbf{w}_i^\top \cdot \mathbf{s}_2^*\ \ \ldots \mathbf{w}_i^\top \cdot \mathbf{s}_N^*]$ (and for the second image, the descriptor vectors are arranged in the form $\mathbf{R} = [\mathbf{r}_1\, \mathbf{r}_2 \ldots \mathbf{r}_M]$), we can compute multiple correlation scores for any orientation, θ_i using a similar expression

$$\mathbf{C}(\theta_i) \quad = \quad \Re\left\{\widehat{\mathbf{S}}_i^{*\top}\mathbf{R}\right\} \tag{4.4}$$

that is simply a matrix multiplication of $\mathcal{O}(M{\times}K{\times}N)$. The matrix $\widehat{\mathbf{S}}_i^{*\top}$ is independent of $\mathbf{R}$ and has to be computed only once for each orientation. Note that we need only the real part of the product, this saves half the computation. We need D times as much storage to store $\widehat{\mathbf{S}}_i^{*\top}$ as we need for $[\mathbf{s}_1^*\, \mathbf{s}_2^* \ldots \mathbf{s}_M^*]$, but the convenient form of (4.4) leads to a significant speedup of the computation (in spite of its higher asymptotic computational complexity). This speedup is mainly due to the speed and simplicity of matrix multiplication as well as faster memory access as the data required for the calculation is stored in contiguous locations in memory and there are very few cache misses. Similar speedups are likely to be available in any programming language that supports efficient matrix manipulation.

We list the timings for the pairwise and anglewise methods in Table 4.2.

[4] In case of 48-point correlation scores, $\mathbf{W}$ is a $48{\times}K$ IFFT operator that includes the appropriate zero-padding

Algorithm 1 Pairwise method

1: **for all** $m = 1$ to M, **do**
2: **for all** $n = 1$ to N, **do**
3: $\mathbf{C}(m, n, \text{all } \theta) = \{\text{IFFT} \sum \mathbf{P}_m \cdot \mathbf{P}_n^*\}$
4: **end for**
5: **end for**

Algorithm 2 Anglewise method

1: $\mathbf{R} = [\mathbf{r}_1 \ \ldots \mathbf{r}_M]$
2: **for all** $i = 1$ to D, **do**
3: $\widehat{\mathbf{S}}_i^* = [\mathbf{w}_i^\top \cdot \mathbf{s}_1^* \ \ldots \mathbf{w}_i^\top \cdot \mathbf{s}_N^*]$
4: $\mathbf{C}(\text{all } m, \text{all } n, \theta_i) = \Re\left\{\widehat{\mathbf{S}}_i^{*\top} \mathbf{R}\right\}$
5: **end for**

These were taken using MATLAB code on a 2.4 GHz Intel CPU with 2GB RAM. The timings are averaged over 10^6 keypoint pairs and several runs of each algorithm were made before taking timings in order to allow optimisation of the FFT routine within MATLAB[5]. The timings given are averages over 10 separate runs.

Method (descriptor size K = 96)	12-point scores D = 12	48-point scores D = 48
Pairwise method	22 μ sec	218 μ sec
Anglewise method	0.6 μ sec	3 μ sec

Table 4.2: Time required to calculate the full matching score for one keypoint-pair, averaged over 10^6 keypoint-pairs.

If speed of calculation is a major consideration, it is possible to compute

[5]MATLAB uses the FFTW [Frigo and Johnson, 2005, Johnson and Frigo, 2008] library to compute FFTs. To explain the concept of a planner, we reproduce the following excerpt from the FFTW documentation: *"First, FFTW's planner 'learns' the fastest way to compute the transform on a particular machine. The planner produces a data structure called a plan that contains this information. Subsequently, the plan is executed to transform the array of input data as dictated by the plan. The plan can be reused as many times as needed. In typical high-performance applications, many transforms of the same size are computed and, consequently, a relatively expensive initialization of this sort is acceptable."*

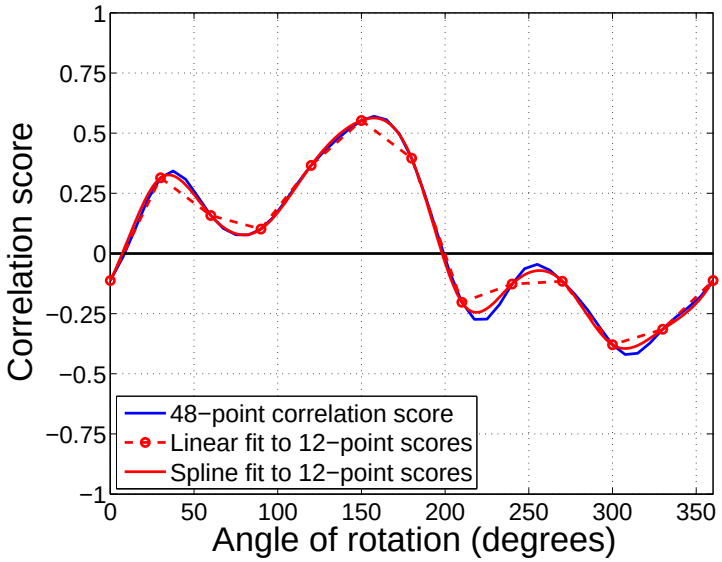


Figure 4.4: 48-point correlation scores (blue) and 12-point correlation scores (red). The 12-point scores are exactly equal to the 48-point scores at relative orientations that are multiples of 30°. Dashed lines show a linear fit through the 12-point scores and solid red curve shows the spline fit through the 12-point scores. The spline fit is not exactly equal to the 48-point curve, but it serves as a very good approximation.

only 12-point cross-correlation scores. The cost of computing one 12-point score is about $\mathcal{O}(12 \log_2 12)$ complex multiply-adds. If 48-point scores are needed, then we can either use an interpolation on the 12-point scores or re-compute 48-point scores for only those keypoint pairs whose maximum scores exceed a certain threshold. We find that 12-point scores do not significantly reduce the matching accuracy and hence suffice in most cases.

The 48-point scores require four times as much computation as the 12-point ones, so theoretically one requires five times more computation for a keypoint pair that passes the initial pruning stage than for one that does not. Assuming that the 12-point score computation requires N_{12} operations and that the fraction of accepted keypoint-pairs is κ, the total computational cost as a fraction of the computation required to compute full 48-point scores for all pairs is $(4\kappa + 1)N_{12}/(4{\times}N_{12}) \approx 0.25 + \kappa$. In practice, the 12 point score computation is more than $4\times$ faster than the 48 point one, as seen in

Table 4.2.

Finally, we note that the conventional FFT-based method (and the zero-padding) for computing correlation scores was proposed primarily for a greater angular resolution (in the relative orientation that is recovered as a result of matching a pair of descriptors (7.5° instead of 30°). If we use 12-point scores in the first stage for speed, one can also consider doing the correlations in the spatial domain (as opposed to the FFT-one), thereby saving the FFT and IFFT steps. This would involve $K/12$ circular correlations of 12-d sequences for 12 shifts. This can also be written in the form of a big matrix product and would probably benefit from the conclusions drawn in Section 4.5.2 *i.e.* large matrix multiplications can be calculated of efficiently if all the requisite data can be accessed in a single memory read operation. The **P**-matrix uses an energy-based normalisation for tolerance to illumination changes. An equivalent normalisation can be applied in the spatial-domain as FFT and IFFT are energy-preserving transforms.

4.6 Matching groups of keypoints

Keypoints often occur together over a range of viewpoints so we also briefly explored the idea of matching groups of them. *Cluster-cluster matching*, our coarse-to-fine keypoint matching approach, is described in [Bendale et al., 2007] (reproduced in Appendix D). The basic idea is to encourage correct matches by enforcing weak spatial inter-match displacement constraints. We first form clusters of keypoints within each image to be matched. Clusters are allowed to overlap one another so that keypoints can contribute to several nearby clusters. All rotations of the pattern of locations of keypoints in a reference cluster are compared with the pattern of location of keypoints in the test cluster by looking for a maximum in the 3D histogram of their matching scores over a range of displacements and rotations $(dx, dy, d\theta)$. The histogram accumulates the matching scores of the keypoint pairs in the candidate clusters. The value of the maximum in the histogram is the matching score for the two clusters and its location gives the approximate relative displacement between the clusters. We used the memory-efficient anglewise al-

gorithm described in Section 4.5.2 to compute the correlation scores. We will not give more details of this method here. The study used an earlier and less reliable version of our keypoint detector and one of the main conclusions was that more reliable individual keypoint matches were required. Given this, we shifted our focus to evaluating individual keypoint matching, as described in Chapters 5–6, and we have not yet updated the cluster-cluster matching study to incorporate our much-improved detector. Other approaches attempting to exploit local spatial layout to aid keypoint matching include [Schmid and Mohr, 1997, Grauman and Darrell, 2005, Mortensen et al., 2005, Leordeanu and Hebert, 2007, Ng and Kingsbury, 2010]. Keypoint locations can also been used to constrain feature correspondences using model selection algorithms [Triggs, 2001]. Traditional approaches for estimating rough geometry between images using keypoint matches are RANSAC [Fischler and Bolles, 1981] and its variants [Chum and Matas, 2005].

4.7 Discussion

In this chapter, we have adapted the Polar matching matrix descriptor for use within the 4S-DTCWT framework. We also formulated the computation of the descriptors and the matching scores in a simple and convenient matrix multiplication form. This leads to significant speedup of these operations, making the system more suitable for use in practical scenarios.

Chapter 5

Evaluation of keypoint detectors and descriptors

In this chapter, we present a new 3D dataset that we created to facilitate evaluation of keypoint detectors and descriptors. This dataset provides the point to point mapping that is used as a reference for the experiments in this chapter. Then we detail the geometry of the setup and the test framework, followed by a quantitative evaluation of our keypoint detector and descriptor alongside other methods. We also present quantitative results for some of the important configurations of the keypoint detector discussed in Chapter 3. Finally, we discuss several issues involved in the evaluation of keypoint detectors and descriptors. The key results are presented in Figures. 5.3 and 5.8.

Throughout this chapter, we shall distinguish between the terms geometric match and appearance match. A *geometric match* of a keypoint is considered to be a match by virtue of being at the correct corresponding location. An *appearance match* of a keypoint is considered to be a match by virtue of being visually similar to it. A geometric match may or may not be visually similar and an appearance match may or may not be at the correct corresponding location. Ideally, we want a match to be both a geometric match as well as an appearance match. We shall also use the term *inlier* interchangeably with the term match.

5.1 3D Dataset

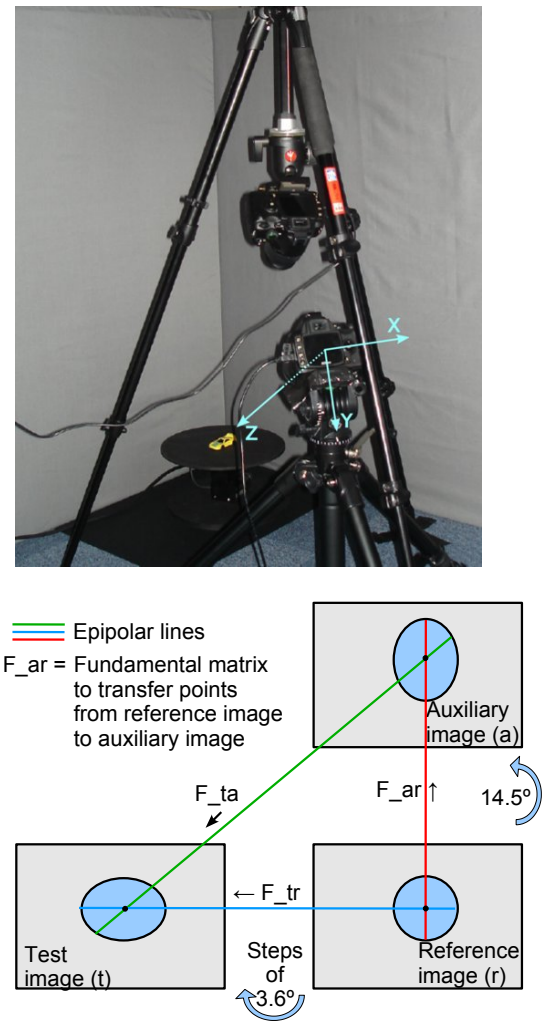


Figure 5.1: *Top*: The rig used to capture the dataset. Note that the upper camera is inverted but this is handled seamlessly by the calibration and evaluation frameworks. If desired, the images may be de-rotated (by 180°) for display purposes. *Bottom*: The three image epipolar matching scheme derived from [Moreels and Perona, 2007].

Previous evaluations of keypoint detectors and descriptors have tended to focus on planar scenes, relying on homographies to generate the ground truth [Mikolajczyk and Schmid, 2005, Schmid and Mohr, 1997]. Evaluations on 3D scenes include [Fraundorfer and Bischof, 2005] (which uses trifocal tensors on

office scenes) and [Moreels and Perona, 2005, 2007] (which uses calibrated images in the form of turn-table sequences). In order to provide an accurate evaluation of methods on real scenes, we wanted to use a method similar to [Moreels and Perona, 2007]. Unfortunately, although these authors provide a robust framework, they have not made their complete ground truth, calibration and test software publicly available. Moreover, the objects typically only occupy a small portion of their images and the image backgrounds are neither plain nor realistic so there are typically many detections on them.

For this reason we created a new dataset containing 3978 calibrated images from 2 cameras of 39 different toy cars on a turn-table with a plain grey background. There are 51 images per car per camera spanning 180°. The images in both PNG and raw (NEF) format, the calibration images and camera parameters including lens distortions, and the MATLAB test scripts are all freely available for download[1]. We used the publicly available DLR CalDe and CalLab toolbox [Strobl et al., 2005] for calibration and DCRAW [Coffin, 2008] for decrypting the NEF (Nikon raw format) files.

Convex polygonal boundaries of the cars in both cameras are available for 14 sequences (first set). Obtaining polygonal boundaries involves extensive manual intervention and has been performed only for the first set. We used the polygonal boundaries to ensure that we only used the detections on the objects in the evaluation. For each car, we use the central 17 of the full 51 views as reference views, so our evaluation is effectively done on $17 \times 14 = 238$ images for each angular separation. We used an approximate bounding box instead of a polygonal boundary for the second and third set but then found that the number of features detected on the cars were too few to allow reasonable evaluation.

Calibration process

A two step calibration process is followed. In the first step, we use the CalDe/CalLab software [Strobl et al., 2005] to obtain the internal parameters of the two cameras as well as the rotation and translation between them. In

[1] http://www-sigproc.eng.cam.ac.uk/imu

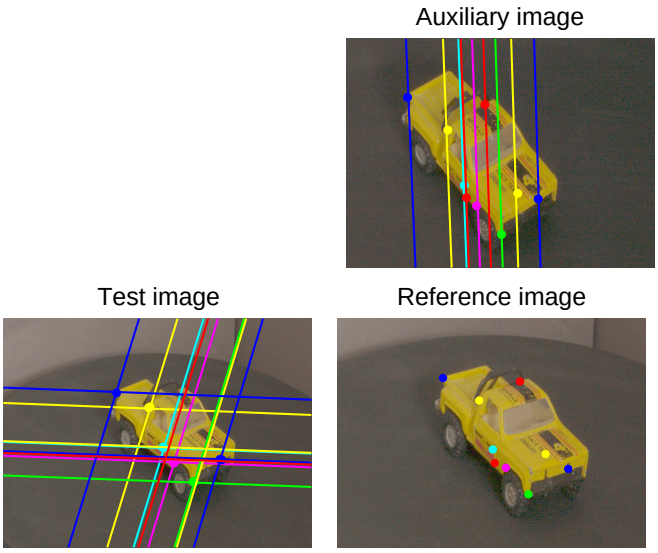


Figure 5.2: Some example images of the set-up used for our dataset and experiments. For each keypoint in the reference image, epipolar constraints and normalised colour cross-correlation are used to estimate the location of the corresponding keypoint in the auxiliary image. The intersection of the reference-image and auxiliary-image epipolar lines in the test image then gives us the predicted location of the corresponding keypoint in the test image (if any).

the second step, we fit an ellipse to the world points obtained by triangulating the points on the edge of the turn-table in images from both the cameras. Points on the edge of the turn-table are selected in one image, an epipolar line is projected in the other image, then the correspondence (intersection of the epipolar line and the edge of the turn-table) is marked manually on the epipolar line. These correspondences are then triangulated using the existing inter-camera calibration to obtain world point coordinates. This has to be done in only one view because the turn-table rotates around its centre, so its boundary is the same in all views. Using the least squares approximation of the true boundary of the turn-table, we obtain its centre and the direction of its axis of rotation. The boundary points are clicked manually. In hindsight, it would have been useful to place a fairly small but obvious marker some distance away from the centre of the turn-table and use the calibration approach described in [Fitzgibbon et al., 1998]. This

involves tracking features across a sequence of images, but no edge detection or manual selection of points is required. Further details of the calibration and data capture processes can be found in [Bendale et al., 2010a].

5.2 Geometry of the test framework

Geometrically, the cameras have similar focal lengths and are at approximately the same distance from the centre of the turn-table, and both look directly at it. One (auxiliary) is placed above and somewhat in front of the other (reference/test) and the difference in elevation is about 15°. This arrangement produces near vertical epipolar lines between corresponding reference and auxiliary images, near horizontal ones between adjacent reference/test images, and similar image scales in all images (which simplifies the evaluation of keypoint scale estimates). For the evaluations, for each 'reference/test' image in turn ('reference'), we use epipolar geometry against its corresponding 'auxiliary' image to find possible matches, then use 3-image epipolar geometry to evaluate whether a corresponding point was found in the designated 'test' image (the reference/test one at a given angular separation from the current 'reference'). Figure 5.1 illustrates the set up. An example can be seen in Figure 5.2.

5.3 Experiments on our 3D Dataset

We tested the BTK keypoint detector against a selection of other methods including the original FKA detector [Fauqueur et al., 2006], the SIFT Difference of Gaussian detector (SIFT-DoG)[Lowe, 2004], and the Intensity Based Region (IBR) [Tuytelaars and Van Gool, 2004], and the Harris-Affine (HAR-AFF) and Hessian-Affine (HES-AFF) detectors [Mikolajczyk et al., 2005]. The most comparable previous evaluations are [Moreels and Perona, 2007] and [Fraundorfer and Bischof, 2005]. To the extent possible, we separated the evaluation of the keypoint detectors from that of the descriptors.

Inlier decision rules

In contrast to previous 3D studies, we base the thresholds used for inlier decisions on the scale of the reference keypoint. Descriptors for coarse scale keypoints use image gradients from coarse scales, hence they can typically tolerate bigger shifts in location of the keypoint. On the other hand, descriptors for fine scale keypoints use image gradients from fine scales, which are resistant to only small shifts in location. To do a realistic evaluation, we need to measure keypoint localisation errors in the way they would affect the descriptors, rather than in absolute terms. Scale-based thresholds are also preferable because when local descriptors are used, they are computed on scale-normalised image patches and localisation errors should be measured in terms of their effect on the descriptors, *i.e.* as a function of the scale of the keypoint. We also ran comparable tests using fixed thresholds – *e.g.* 5 pixels from the epipolar intersection, irrespective of keypoint scale. This did not cause any noticeable change in the shape or relative ranking of the curves (see Figure 5.4). The fixed-threshold scheme is biased towards fine scale keypoints in the sense that it allows greater relative localisation errors for these than for coarse scale ones. Fine scale keypoints are usually more unstable than coarse scale ones.

Gamma compression

We find that gamma compression of the image – $\mathbf{I} \leftarrow [\mathbf{I} + c]^{\gamma}$ with $c \sim 20$–30 and $\gamma \sim 0.3$–0.5 – improves the performance of all of the keypoint detectors except SIFT-DoG (which has built-in illumination invariance). We use this correction in all of the tests.

5.4 Detector repeatability

Figure 5.3 summarises the results of an evaluation of the repeatability of various keypoint detectors under changes in viewpoint, using scale-dependent thresholds. The test range is from $-57.6°$ to $57.6°$ in steps of $3.6°$ and excluding $0°$. Only reference and auxiliary images from the central $60°$ of the $180°$

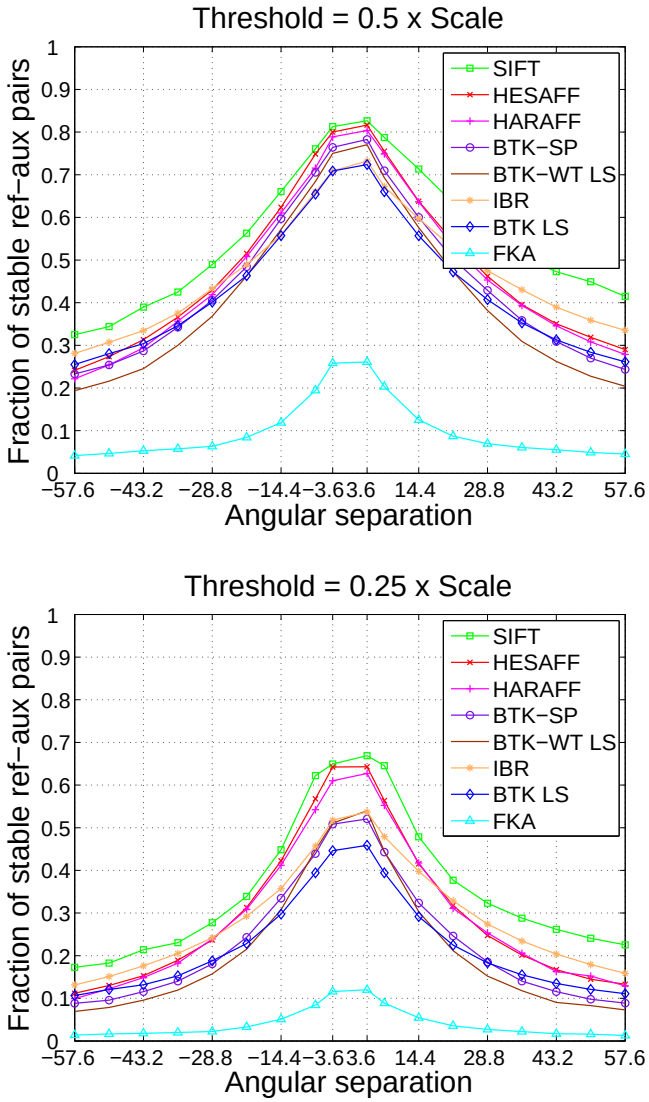


Figure 5.3: The repeatability of various keypoint detectors under changes in viewpoint. We plot the fraction of reference–auxiliary pairs that have a test-image keypoint at the estimated location, for a range of angular separations and for two acceptance thresholds. A gradually falling curve indicates good tolerance to changes of viewpoint, but the curves become more peaked as the constraint on localisation error is tightened. The BTK detector improves significantly on FKA owing to its better position and scale estimation. Its performance is now in line with the other established detectors. All of the detectors here are set to find about 100 ± 5 reference–auxiliary pairs per image tested. Markers are shown at steps of 7.2° with extra points at 3.6° for better resolution in the critical region near zero.

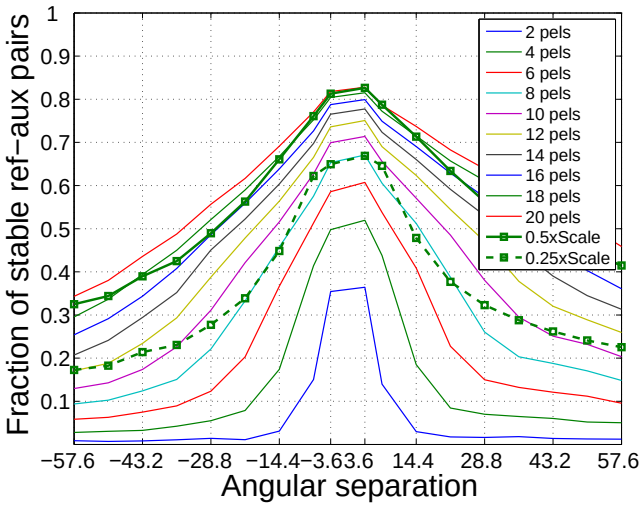


Figure 5.4: Use of scale-dependent vs scale independent thresholds. The set of solid curves show the results obtained as per the experiment shown in Figure 5.3 for SIFT-DoG keypoints with a fixed threshold (irrespective of their scale) of 2-20 pixels for all keypoints. The green lines with markers show the results for thresholds 0.5×Scale (solid) and 0.25×Scale (dashed).

sequences are used so that each ref–aux pair will have the full ±57.6° range of test viewpoints. The thresholds are set so that on average, each detector finds 100 ± 5 ref–aux keypoint pairs per image tested. For each viewing angle, we measure the probability that the detector re-detects a keypoint at the corresponding physical location and scale on the object. To do this, we locate the auxiliary-image point corresponding to a given reference point by searching for the best match using normalised colour cross-correlation among the keypoints within a certain scale-independent distance of the reference point's epipolar line in the auxiliary image. The selected ref–aux pairs are used to predict keypoint positions in the test image by epipolar line intersection, and if a keypoint is detected within a certain distance of this position – 0.5× the scale of the reference keypoint for Figure 5.3 (top) and 0.25× the scale for Figure 5.3 (bottom) – we consider it to be a successful detection. Unsurprisingly, Harris-Affine, Hessian-Affine and SIFT-DoG detectors show very similar performance: All these detectors are based on Difference-of-Gaussian scale-space. Note the extent to which the BTK detector improves on the

FKA one, owing to its better position and scale estimation.

5.4.1 Comparison of various scale estimation methods

We now test the repeatability (with respect to changes in viewpoint) of a selection of different keypoint detection methods within the 4S-DTCWT scale-space framework. The variants compared are

No Subpixel. 2D spatial maxima are detected in each level without any 2D spatial sub-pixel refinement. Each keypoint is checked to ensure it attains a maximum in scale.

No Sublevel. 2D spatial maxima are detected in each level followed by 2D spatial sub-pixel refinement. Each keypoint is checked to ensure it attains a maximum in scale.

Spline. 2D subpixel spatial maxima as per *No Sublevel* followed by a 1-D spline interpolation for sub-level scale refinement (Section 3.7.2). Each keypoint is checked to ensure it attains a maximum in scale.

LS. Least Squares quadratic surface fitting (Section 3.6.2).

Wt LS. Least Squares quadratic surface fitting with Gaussian weighting based on the distance from the centre of the 3×3×3 region (Section 3.7.1).

Half Max NMS: Half maximum scale estimation with non-maximal suppression (Section 3.5.2).

DN $\sigma = 0.25$. Damped Newton method with $(\sigma_x, \sigma_y) = 0.25\times$Scale, 3 levels (detection level and 1 above and below each) are used (Section 3.6.1).

DN $\sigma = 0.5$. Damped Newton method with $(\sigma_x, \sigma_y) = 0.5\times$Scale, 3 levels (detection level and 1 above and below each) are used (Section 3.6.1).

Figure 5.5 presents the results of an experiment similar to that in Figure 5.3, from which we conclude

- When the 2D spatial maxima detected at individual levels are refined to provide sub-pixel position estimates, this improves the results.
- Half-Maximum scale estimation tends to worsen the results relative to using just the sampling interval of the detection level as the scale estimate.

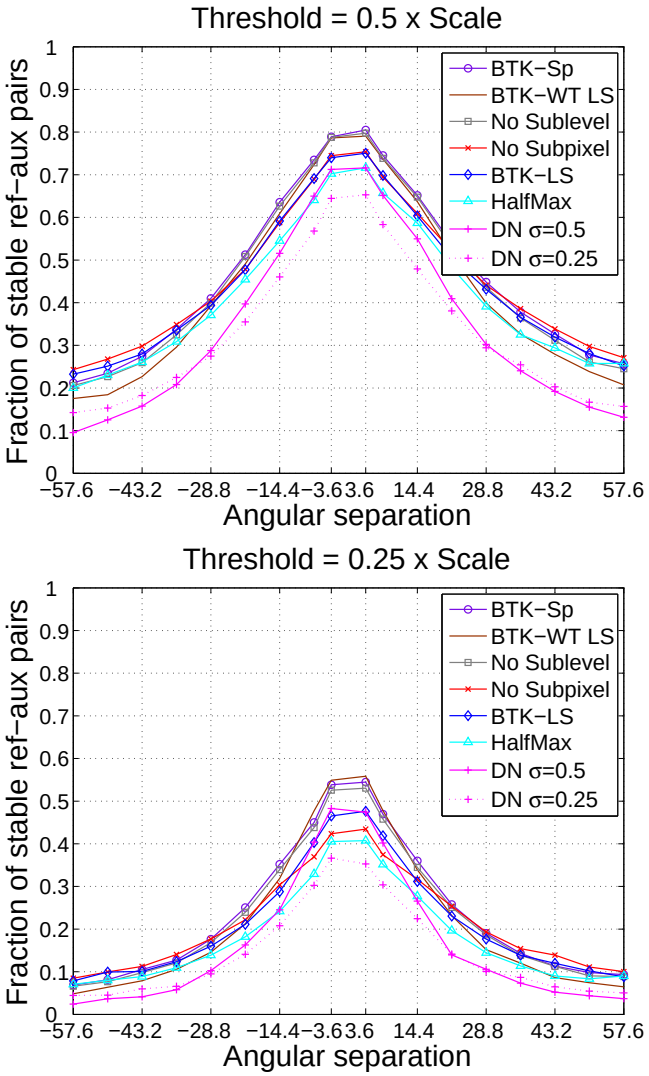


Figure 5.5: The repeatability of several subpixel and sub-level keypoint location estimators for our DTCWT based detector under changes in viewpoint. We plot the fraction of reference–auxiliary pairs that have a test-image keypoint of the estimated scale (radius) at the estimated location, for a range of angular separations and for two acceptance thresholds. The experimental setup is same as in Figure 5.3, except that this experiment uses only a subset of the dataset (cars 01, 02, 04). All of the methods have the same input keypoint response, so the differences in performance are solely due to the scale estimation method.

- Using a larger smoothing kernel in Damped Newton tends to merge more points into one, leaving fewer but better points. At small angular separations this has a desirable effect, but at large angular separation it seems that fewer points are re-detected.
- An unweighted least squares surface fit is a poor model of the actual peak keypoint responses so Least Squares tends to drag keypoints away from their true position and scale, leading to a worse result than not doing any scale refinement.
- The spline-fit method, which consists of taking a 2D spatial maximum that is a local maximum in scale as well, and then performing spline based 1-D interpolation in scale through its location performs very well.
- If Gaussian weighting is used such that the centre of the 3×3×3 grid has more weight than the corner points of the grid, the least squares fit of the centre point improves, leading to very good overall results.

Both the spline fit and Gaussian weighted least squares methods give very good results. The slight differences are due to the fact that the latter refines the position estimate as well as refining the scale estimate. Gaussian weighted least squares tends to produce fewer keypoints that the spline fit method because it enforces a strict maximum over a 3D region, whereas the spline fit method enforces a maximum only over the three central axes of this 3D region.

5.4.2 Points with multiple orientations

In the SIFT system, an orientation is associated with every SIFT-DoG keypoint. The orientation estimate is based on an orientation histogram of the local image gradients. If there are multiple dominant peaks in the orientation histogram (dominant means that the peak value is at least 80% as high as the highest peak), then separate keypoints are output corresponding to each peak in the orientation histogram. These keypoints have the same spatial location and scale but they differ in orientation. The descriptor is then formed by rotating the image patch so as to cancel the effect of the detected

orientation, in order to make the extracted descriptor invariant to rotational transformations.

This is appropriate when SIFT-DoG keypoints are used with SIFT descriptors (or any other descriptor that is not rotation-invariant). However, if these keypoints are used with a rotation-invariant descriptor, then all such keypoints (with the same x, y, s but different orientations), lead to the same descriptor vector. In other words, whether or not these detections should be treated as multiple keypoints depends on the descriptor that they will be used with and evaluation results presented here differ in the two cases. As SIFT-DoG detector is a popular keypoint detector that is used in a variety of applications, albeit not always with the SIFT descriptor, we evaluated its repeatability with respect to changes in viewpoint in the two cases

i) in which keypoints with the same (x, y, scale) but different orientations are considered to be a *single* detection (*i.e.* the multiple orientations are viewed as part of the descriptor matching process); and

ii) when different orientations are considered to be *different* detections.

Figure 5.6 shows the results of this evaluation. The green $\square$'s show the results for Case 1 (single detection) and the blue $\triangle$'s show the results for Case 2 (multiple detections). The experimental setup is the same as in Figure 5.3. We see that the performance is slightly different in the two cases, particularly for large angles. A fraction of the (geometric) matches have the correct (x, y, scale) but the wrong orientation, particularly for large angles.

5.4.3 Simultaneous stability in position and scale

So far, we have not used the scale of the test keypoint anywhere in the evaluation. For two keypoint descriptors to match reliably, repeatability of scale is just as important as repeatability of position. Figure 5.7 shows the result of an evaluation that accepts a test keypoint as being a valid match for a reference keypoint only if the error in scale is less than half an octave as well as requiring the position error to be less than a certain threshold.

The markedly different results of Figures 5.7 and 5.3 clearly illustrate the need to consider the constraints on both scale and position when evaluating

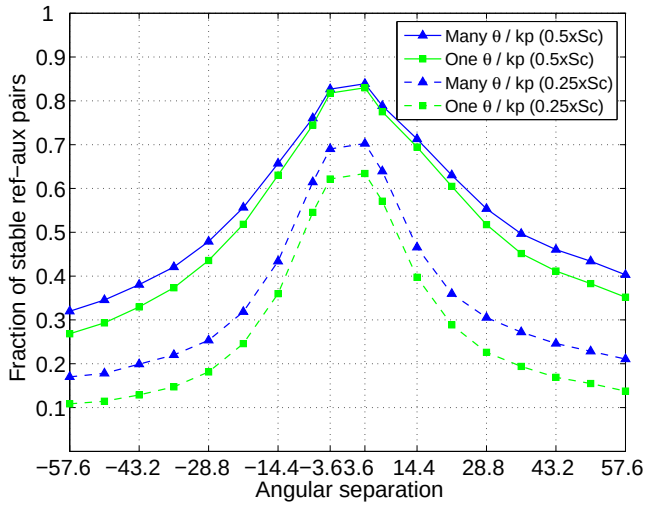


Figure 5.6: SIFT-DoG detector repeatability for keypoints with multiple orientations. Proceeding as in Figure 5.3, we see that the performance is slightly different in the two cases. The green curves are obtained by replacing all keypoints with the same (x, y, scale) but different orientations by a single keypoint. The curves are plotted for threshold $= 0.5\times$Scale and $0.25\times$Scale. SIFT-DoG keypoints that are equivalent in (x, y, scale) but multiple orientations may be counted as a single detection (green $\square$'s) or as multiple detections (blue $\triangle$'s) for the purposes of evaluation of the keypoint detector. The difference between the two approaches becomes significant at tighter thresholds.

keypoint detectors. For example, due to the additional requirement of scale matching, the accuracy at small angles drops from approximately 68% to 62% for SIFT (localisation error less than $0.25\times$ Scale). Note that SIFT demonstrates greater scale accuracy because it fires primarily on blob-like features that are very well localised in scale.

5.5 Descriptor repeatability

For each viewing angle, we measure the likelihood of a good matching score (and hence a good matching rank) for the descriptors corresponding to points that are known to be geometric inliers. For each reference point, we first determine which keypoints in the test image are geometric inliers (*i.e.* are

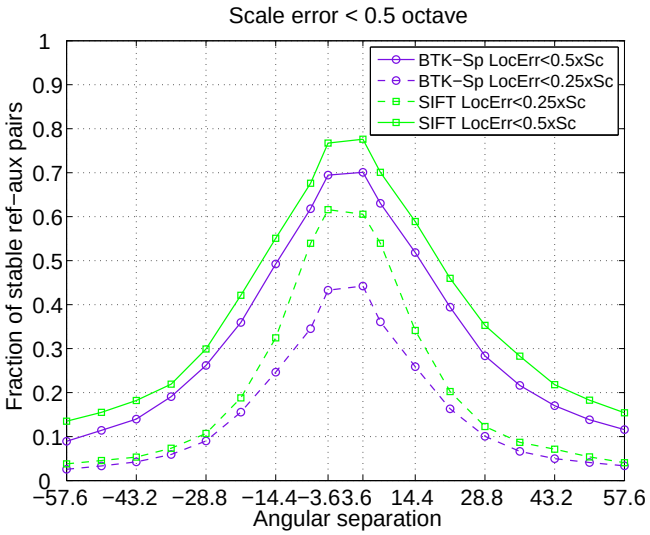


Figure 5.7: Detector repeatability: Scale and position. We proceed as in Figure 5.3, but use two constraints to test keypoints in the test image. The test keypoint should have a spatial localisation error of at most 0.5×Scale of reference keypoint (or 0.25×Scale for the dotted lines) and a scale error of at most half an octave. Note that simultaneous stability in position and scale is significantly harder to achieve than stability in position only.

located within a certain distance from the intersection of the epipolar lines). If the reference point has at least one geometric match, we select the rank of that keypoint which lies within the geometric match region and achieves the highest rank in appearance-based descriptor matching of all keypoints within the test image with the reference keypoint's descriptor. This process finds the highest rank of a descriptor match but with the additional constraint that the keypoint must also be a geometric match.

Figure 5.8 summarises the result of such an evaluation of the SIFT and BTK descriptors. If the descriptor similarity remains roughly constant as the viewing angle increases, the distribution of ranks should also remain roughly constant because there are about the same overall number of detections in each image, whereas if the similarity degrades, the rank should increase with angle. To quantify this, for each descriptor and keypoint type, we plot the histogram of its ranks for each angle. Figure 5.8 shows that – conditioned on the associated point being detected at all – the descriptor similarity is

indeed roughly constant with angle.

Comments. For display purposes, each histogram (column) is normalised to sum 1. Although the overall number of detections remains roughly constant with angle, the number of detections that are geometric inliers decreases with increasing angle as we saw in Figure 5.3 (*i.e.* on average, old points that are no longer detected are replaced with new and different ones). Hence different inlier histograms have similar distributions but very different numbers of counts. The epipolar geometry of our 3-image setup is weaker at large angles owing to shallower intersection angles between epipolar lines. This produces a wider search region for matches at these angles and in turn, slightly decreases the observed ranks at large angles, where more candidates are tested. This effect is purely geometric and not due to the descriptor performance. The slight asymmetry is probably due to lighting coming mainly from the right.

Efficiency

On a 1536×1024 image with 389 keypoints, our current MATLAB implementation takes 11.5 seconds: 6.3 to evaluate the 4S-DTCWT pyramid; 4 to detect keypoints; and 1 to compute their descriptors. A C implementation would probably be at least twice as fast. In comparison, Lowe's C implementation of SIFT [Lowe, 2004] takes 5.5 seconds on the same image for 329 SIFT keypoints. In both cases the runtime is roughly linear in the total number of pixels in the image. The Harris-Affine implementation [Mikolajczyk, 2005, Mikolajczyk and Schmid, 2004] took 6.7 seconds and produced 239 features. The Hessian-Affine implementation [Mikolajczyk, 2005, Mikolajczyk and Schmid, 2004] took 4.4 seconds and produced 379 features.

5.6 Summary and future work

We have presented an evaluation of the DTCWT based keypoint detector and descriptor developed in Chapters 3 and 4. The performances of the detector and descriptor are now in line with those of other established methods. The new scale space pyramid 4S-DTCWT leads to a better coverage of key-

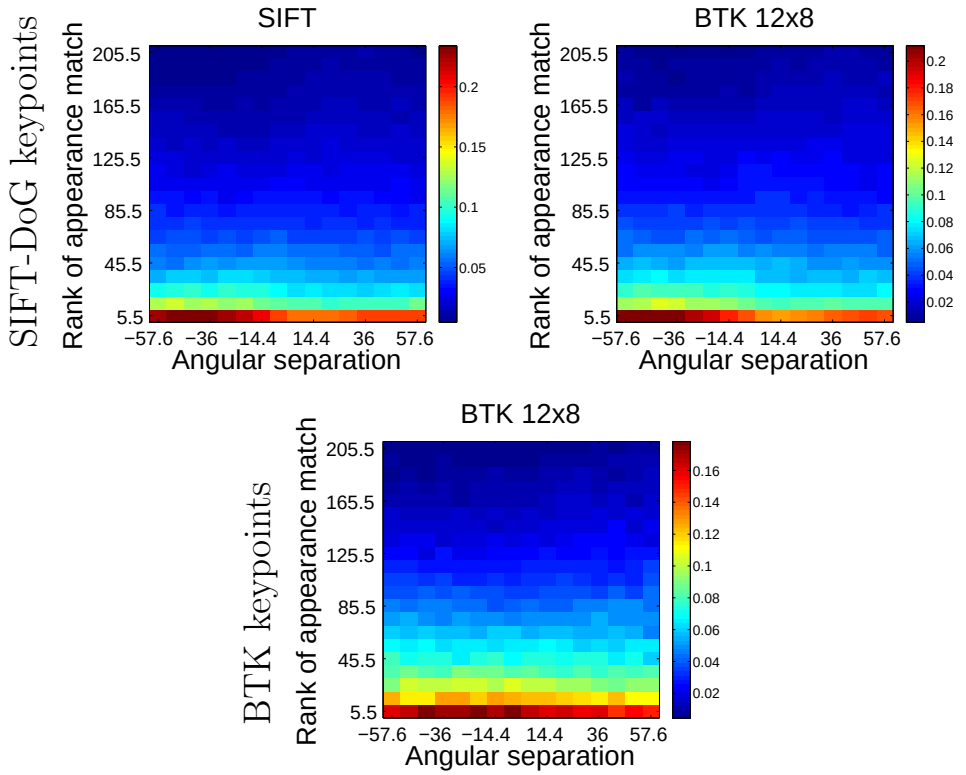


Figure 5.8: Normalised histograms of the rank of the true-correspondence keypoint among all of the keypoints in the image, where rank is measured by inter-descriptor distance to the target descriptor. *Top*: Histograms for SIFT and BTK descriptors over SIFT-DoG keypoints are plotted as a function of the inter-image angle. Warmer colors indicate higher frequencies. The figure shows that (unlike detectors) increasing angular separation has relatively little effect on these descriptors. *Bottom*: BTK descriptors are computed on BTK keypoints in bottom row to facilitate comparison.

points across the range of scales, *i.e.* there are fewer missed detections. The BTK detector demonstrates significant improvement in scale and spatial localisation compared to the FKA detector. In comparison with other feature detectors, we see that

- **Repeatability.** BTK features are almost as repeatable as other leading feature detectors. The stability of the BTK features is significantly better than that of its predecessor FKA. The 4S-DTCWT scale space offers finer scale sampling, leading to more stable scale estimates and

also better spatial localisation. Whilst the overall stability is still not quite as good as that of SIFT, other merits of the detector might be a good reason to tolerate this slight reduction of stability. While improving the spatial localisation of the BTK detector might be possible, improving the scale localisation might prove to be tricky (and measuring the scale stability is even trickier). The reason for this is that BTK detects a wide range of features such as corners and junctions that do not inherently have a scale and in most cases, the scale associated with such features is due to the contribution of nearby features.

- **Complementarity.** BTK detector detects a variety of 2D features - corners, blobs and junctions. Hence, BTK detects different features as compared to those detected by other feature detectors (based on Gaussian scale space), because it uses a different response function as it's base. When used alongside another feature detector, it will potentially provide a better coverage of the image than any single feature detector may on its own. In most computer vision tasks, feature detection and matching is used as the first stage. The fraction of stable features (those that can be re-detected and matched correctly) matters, but the total number of features is also very important. In a computer vision system, having stable features is good, having a greater number of stable features is better, especially if they cover all places of likely interest in the image. Thus BTK can be usefully combined with other feature detectors leading to improved overall performance (by merit of being able to detect sufficiently different features).

- **Improved orientation handling.** The BTK descriptor features a novel orientation handling mechanism. This allows us to match descriptor not only at their best relative orientation, but also at other orientations, leading to a smooth matching process which is useful in applications like image registration. Thus the matching process provides a two-fold ranking as it's output, telling us how well two points match each other over a range of relative orientations, without the need to store multiple copies of each descriptor. This technique of handling

orientation removes the need for the detector to estimate an orientation for each interest point which can be difficult (for points with directional symmetries) and unreliable.

We also introduced a new dataset and explored several issues in evaluation methodologies. In the future, it may be interesting to investigate the following aspects:

- We had to set all the detectors to find about 100 ± 5 reference–auxiliary pairs per image tested. We chose this number because SIFT produced 100 ± 5 detections on average and SIFT was the only detector that we did not have any control over[2]. It was important to have roughly the same number of features for all detectors because our evaluations of keypoint detectors are somewhat sensitive to the number of features detected. For example, if a detector detects a great many features, the probability that a keypoint is present close to the intersection of the epipolar lines is increased and the current evaluation framework does not have any reweighting to account for such variations so it might be interesting to validate our results at different detection thresholds. Recently, an open source implementation of SIFT has become available [Vedaldi and Fulkerson, 2008], which allows better control.

- Another point to note is that while repeatability is one useful measure of how good a detector is, it is not the only one. Other aspects such as the spatial spread of keypoints, the spread in detection scales, and the distinctiveness (uniqueness) of the detected keypoints are not measured by evaluations based on repeatability. Nevertheless, repeatability has been used widely as the basis of evaluating keypoint detectors, hence it is a reliable tool for comparisons.

- It is now well accepted that the re-localisation accuracy of keypoints (or affine regions) is reduced as a result of changes in viewpoint. However, there has not been much work done to ascertain how much of this loss

[2]We used D. Lowe's publicly available binary, which does not give any control over the parameters.

is a result of other effects such as pixelation, resampling, lens distortion, blurring, exposure *etc.*, and how much is a result of the detection method.

Even though many studies have been done in this area, it continues to be an active area of research. The sheer variability of the data and the multitude of choices one is faced with while making rules for evaluation of keypoint methods makes this an interesting and difficult problem. We presented some new ways of evaluating keypoint methods. Having established that the performance of the keypoint detector and descriptors we developed in chapters 3 and 4 is now comparable to that of other methods, we shift focus to a new geometry based evaluation framework to help us overcome some of the problems we faced while doing the current evaluation. This framework is described in the next chapter.

Chapter 6

Multiscale epipolar geometry

So far our evaluation of keypoint detectors (*c.f.* Chapter 5) has treated keypoints as points, measuring only uncertainties in spatial location. In practice, recent detectors feed a scale as well as a spatial location to a keypoint descriptor. Corresponding keypoints must correspond in both scale and spatial location. In this chapter, we give an **enhanced epipolar constraint** that **exploits both positions and scales**.

Our method is a *purely geometric* way of evaluating *multiscale* keypoint detections. Multiscale keypoints are treated as ellipses that must be matched across images - for example this happens if they are projections of a (possibly planar) 3D ellipsoid on different image planes. In this chapter, the epipolar geometry is assumed to be known (at least approximately). The method defines a distance metric between ellipses from different images that are being matched. This measure corresponds to the disparities in the epipolar matching of position and scale.

In Section 6.2, we re-derive the main result from [Triggs and Bendale, 2010] for the sake of completeness. The theory there was developed by Bill Triggs[1]. Some aspects of this chapter were published in [Triggs and Bendale, 2010] (which is included in Appendix B). We then apply the framework to

[1]The problem was identified while the authors were working on evaluation of keypoint detectors for [Bendale et al., 2010b]. Bill Triggs was responsible for developing and implementing the basic method and writing the theoretical part of the paper. Pashmina Bendale was responsible for implementing the test framework, running the tests, analysing the results and writing the experimental part of the paper.

the evaluation of keypoint detectors. The reader is directed to Appendix A for fundamentals of epipolar geometry and to [Hartley and Zisserman, 2003] for an in-depth coverage of this topic.

We introduce the problem in Section 6.1 and give a brief derivation of the method in Section 6.2. We present experimental results in Section 6.3 and conclude in Section 6.5.

6.1 Motivation

Many recent keypoint detectors [Bendale et al., 2010b, Lowe, 2004, Mikolajczyk and Schmid, 2004, Triggs, 2004] associate a local scale (for multiscale detectors) or even a full affine frame (for affine-invariant detectors) to each detected keypoint. The conventional epipolar constraint [Hartley and Zisserman, 2003] is a powerful tool for matching keypoints (salient local features) between pairs of images, but in its standard form it treats the detected keypoints as point-like entities without intrinsic scales.

Previous work on evaluation of keypoint detectors includes (amongst others) [Mikolajczyk et al., 2005, Moreels and Perona, 2007] and [Fraundorfer and Bischof, 2005]. While [Mikolajczyk et al., 2005] used homography-based mapping on planar scenes and considered scale mismatch to some extent, this study did not consider a full 3D environment. Both [Moreels and Perona, 2007] and [Fraundorfer and Bischof, 2005] were set in full 3D environments, but [Moreels and Perona, 2007] used appearance based cues (descriptors based on scale-normalised patches) for correspondence search. While the approach in [Fraundorfer and Bischof, 2005] used trifocal tensor and depth maps to generate ground truth, and separated the evaluation of planar and non-planar parts of the images, they still used appearance-based cues for correspondence search. Furthermore, none of these studies (except [Mikolajczyk et al., 2005] for the 2D case) explicitly measures scale-mismatch as part of evaluation of the keypoint detector (only implicitly as part of the descriptor).

Prior work on matching blobs using epipolar geometry includes [Forssén and Moe, 2004]. They detect coloured blobs in (mainly planar) images, estimate an approximate homography using RANSAC and subsequently use

epipolar geometry to project the ellipses (and their shape) to the other image, but do not define an error metric – the errors between the ellipses projected from first image onto the second image and the ellipses detected in the second image have to be assessed visually[2]. Our method is based on the well-known 'Kruppa constraints' for correspondence[3] between conics [Porrill and Pollard, 1991, Faugeras et al., 1992, Kahl and Heyden, 1998, Hartley and Zisserman, 2003]. Scale mismatch for keypoints has not been seen in this light before.

Before beginning, we note the following about the method presented in this section:

- This method is purely geometric. It provides a framework for deciding whether a keypoint is re-detected in the correct location with the correct scale without using visual descriptors. There is no consideration of the detailed image content within the keypoint regions and we make no attempt to create a detailed point to point matching of them.
- It is assumed that the epipolar geometry (Fundamental matrix) is known.

In the following sections, we present an enhanced epipolar constraint that exploits both positions and scales, thus making correspondence search 2-4 times more accurate in practice.

6.2 Brief derivation

Let $\mathbf{F}$ be the fundamental matrix such that $\mathbf{x}^\top \mathbf{F}\, \mathbf{x}' = 0$ is the epipolar constraint between left image point $\mathbf{x}$ and right image point $\mathbf{x}'$. The Singular Value Decomposition (SVD) of $\mathbf{F}$ can be written as

$$\mathbf{F} \;=\; \mathbf{U}\,\mathbf{S}\,\mathbf{V}^\top \;=\; \begin{pmatrix} -\mathbf{v} & \mathbf{u} & \mathbf{e} \end{pmatrix} \begin{pmatrix} \mu & & \\ & \nu & \\ & & 0 \end{pmatrix} \begin{pmatrix} \mathbf{u}' & \mathbf{v}' & \mathbf{e}' \end{pmatrix}^\top \;=\; \mathbf{u}\,\nu\,\mathbf{v}'^\top - \mathbf{v}\,\mu\,\mathbf{u}'^\top \quad (6.1)$$

[2]Some work on measuring distances between corresponding epipolar tangents has been presented in the associated technical report by the same authors (*c.f.* [Forssén and Moe, 2004])

[3]Kruppa equations are an algebraic representation of the conic correspondence of epipolar lines tangent to a conic - see Section 19.4 in [Hartley and Zisserman, 2003] for more details.

where the pairs $(\mathbf{u}, \mathbf{u}')$ and $(\mathbf{v}, \mathbf{v}')$ are in epipolar correspondence: $\mathbf{u}^\top \mathbf{F}\, \mathbf{u}' = 0 = \mathbf{v}^\top \mathbf{F}\, \mathbf{v}'$. The epipoles $\mathbf{e}$ and $\mathbf{e}'$ can be expressed in 3×3 skew cross product matrix form as

$$[\,\mathbf{e}\,]_\times = \mathbf{u}\,\mathbf{v}^\top - \mathbf{v}\,\mathbf{u}^\top \quad \text{and} \tag{6.2}$$

$$[\,\mathbf{e}\,]'_\times = \mathbf{u}'\,\mathbf{v}'^\top - \mathbf{v}'\,\mathbf{u}'^\top \quad . \tag{6.3}$$

Next, we define 2×3 epipolar pencil projection matrices as

$$\mathbf{B} \equiv \begin{pmatrix} \mathbf{u} & \mathbf{v} \end{pmatrix}^\top, \qquad \mathbf{B}' \equiv \begin{pmatrix} \mu\,\mathbf{u}' & \nu\,\mathbf{v}' \end{pmatrix}^\top \tag{6.4}$$

with $\mathbf{B}\,\mathbf{B}^\top = \mathbf{I}_{2\times 2}$. The matrices $\mathbf{B}, \mathbf{B}'$ project image points onto the epipolar pencil, as illustrated in Figure 6.1.

It follows from (6.4) and (6.1) that

$$\mathbf{F} = \mathbf{B}^\top \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \mathbf{B}' \quad \text{and} \tag{6.5}$$

$$[\,\mathbf{e}\,]_\times = \mathbf{B}^\top \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \mathbf{B} \quad . \tag{6.6}$$

Pairs of image points are in epipolar correspondence iff their epipolar pencil projections coincide: $\mathbf{B}\,\mathbf{x} \sim \mathbf{B}'\,\mathbf{x}'$. Multiscale keypoints can be represented as ellipses and written as 3×3 symmetric dual-form conic matrices $\mathbf{q}$, $\mathbf{q}'$. Two keypoint ellipses $\mathbf{q}, \mathbf{q}'$ are in epipolar correspondence iff the pair of epipolar lines tangent to $\mathbf{q}$ is in correspondence with the pair from $\mathbf{q}'$. Conic correspondence is encapsulated by the 'Kruppa constraints' [Porrill and Pollard, 1991, Faugeras et al., 1992, Hartley and Zisserman, 2003]

$$[\,\mathbf{e}\,]_\times\ \mathbf{q}\ [\,\mathbf{e}\,]_\times^\top \ \sim\ \mathbf{F}\,\mathbf{q}'\,\mathbf{F}^\top \tag{6.7}$$

Substituting (6.5)-(6.6) in (6.7), the constraint on conic correspondence reduces to

$$\mathbf{B}\,\mathbf{q}\,\mathbf{B}^\top \ \sim\ \mathbf{B}'\,\mathbf{q}'\,\mathbf{B}'^\top, \tag{6.8}$$

which happens iff the epipolar pencil projection matrices of $\mathbf{q}, \mathbf{q}'$ coincide up to scale. Evaluating the dual conic $\mathbf{q}$ on the epipolar line vector corresponding

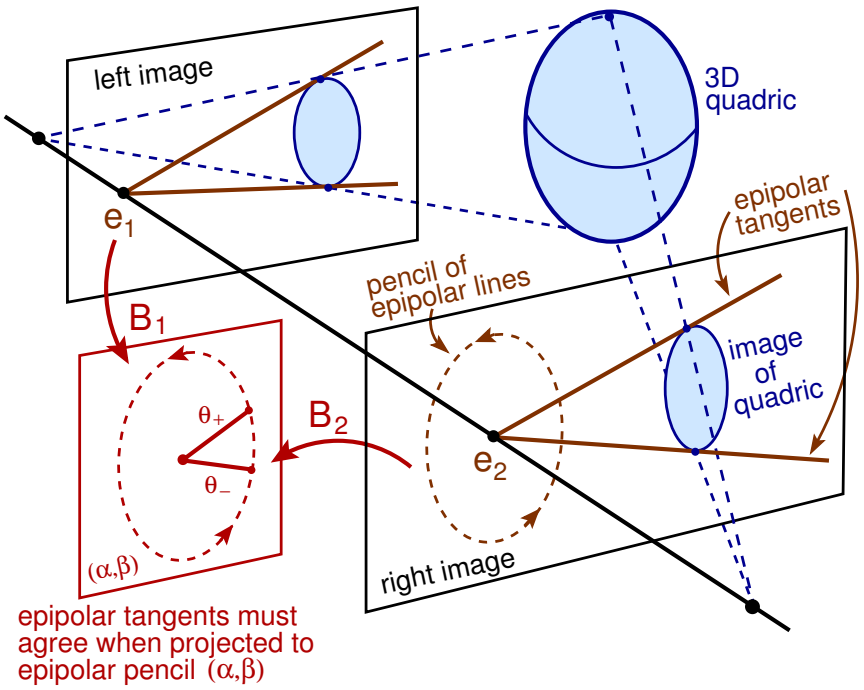


Figure 6.1: **Elements of conic correspondence.** A 3D quadric projects to two image conics, whose pairs of tangent epipolar lines are also in correspondence. If the conics or line pairs are projected onto the epipolar pencil using the 2×3 epipolar projectors $\mathbf{B}, \mathbf{B}'$, the Reduced Kruppa Constraints state that the two projections agree [Porrill and Pollard, 1991]. Figure taken from [Triggs and Bendale, 2010].

to 'point' $(\alpha, \beta)^\top$ induces a corresponding reduced quadratic form on the pencil

$$\begin{pmatrix} \beta & -\alpha \end{pmatrix} \mathbf{B}\,\mathbf{q}\,\mathbf{B}^\top \begin{pmatrix} \beta \\ -\alpha \end{pmatrix} \;=\; a\,\alpha^2 - 2\,b\,\alpha\beta + c\beta^2 \tag{6.9}$$

Representing (α, β) in parametric form as $(\alpha, \beta)^\top \sim (\cos\theta, \sin\theta)^\top$, and letting the roots of (6.9) be $\theta_\pm = \bar{\theta} \pm \delta\theta$ ($\delta\theta$ is small), the RHS of (6.8) must be proportional to $(\theta - \theta_+)(\theta - \theta_-)$ and hence to

$$\begin{aligned} \sin(\theta-\theta_+)\sin(\theta-\theta_-) &= (\sin\theta\cos\theta_+ - \cos\theta\sin\theta_+)(\sin\theta\cos\theta_- - \cos\theta\sin\theta_-) \\ &= \alpha^2 \sin\theta_+ \sin\theta_- - \alpha\beta\sin(\theta_+ + \theta_-) + \beta^2 \cos\theta_+ \cos\theta_- \,. \end{aligned} \tag{6.10}$$

Comparing (6.9) and (6.10), the values of (a, b, c) can be read off as (up to scale)

$$\begin{pmatrix} a \\ b \\ c \end{pmatrix} \sim \begin{pmatrix} \sin\theta_+ \sin\theta_- \\ (1/2)\sin(\theta_+ + \theta_-) \\ \cos\theta_+ \cos\theta_- \end{pmatrix} = \frac{1}{2}\begin{pmatrix} \cos 2\delta\theta - \cos 2\bar{\theta} \\ \sin 2\bar{\theta} \\ \cos 2\delta\theta + \cos 2\bar{\theta} \end{pmatrix} \tag{6.11}$$

$$\begin{pmatrix} p \\ q \\ r \end{pmatrix} \equiv \begin{pmatrix} c-a \\ 2b \\ c+a \end{pmatrix} \sim \begin{pmatrix} \cos 2\bar{\theta} \\ \sin 2\bar{\theta} \\ \cos 2\delta\theta \end{pmatrix} \tag{6.12}$$

Using scaling of $p^2 + q^2 = 1$ and assuming that both the position uncertainty and the scale uncertainty of a typical keypoint are proportional to its scale and that deviations are approximately Gaussian, we introduce two normalised distances that measure respectively the normalised ellipse position mismatch and the normalised ellipse scale mismatch in terms of the pencil projection coordinates of the two ellipses,

$$d_{\bar{\theta}} \equiv \frac{\sin^2 2(\bar{\theta}-\bar{\theta}')}{\sin^2\delta\theta + \sin^2\delta\theta'} = \frac{(p\,q' - q\,p')^2}{1-(r+r')/2} \tag{6.13}$$

$$d_{\delta\theta} \equiv \left(\frac{\sin\delta\theta}{\sin\delta\theta'}\right)^k + \left(\frac{\sin\delta\theta'}{\sin\delta\theta}\right)^k - 2 = \left(\frac{1-r}{1-r'}\right)^{k/2} + \left(\frac{1-r'}{1-r}\right)^{k/2} - 2 \tag{6.14}$$

Here $(\bar{\theta}, \delta\theta)$, $(\bar{\theta}', \delta\theta')$ are corresponding mean angle, angular width coordinates on the (projective) epipolar pencil. (6.13), (6.14) have appropriate small and large angle limits and embody a statistical model where keypoint location and scale uncertainties are proportional to keypoint scale.

Summary of the method. For each ellipse, the matrix $\mathbf{q}$ or $\mathbf{q}'$ is projected using $\mathbf{B}$ or $\mathbf{B}'$ (from the SVD of $\mathbf{F}$) to obtain $(a, b, c)^\top$ (6.8), normalized to give $(p, q, r)^\top$ (6.12). 'Distances' between these vectors are then computed using a weighted sum of (6.13) and (6.14) and used to decide whether pairs of ellipses might correspond.

The method can be summarised as follows:

- Represent multiscale keypoints as image ellipses $\mathbf{q}$ or $\mathbf{q}'$;
- Invoke the 'Kruppa constraints' that link corresponding ellipses (6.7);
- Project to the "epipolar pencil" (the 1-D family of epipolar lines) using $\mathbf{B}$ or $\mathbf{B}'$ to get reduced constraints linking 1-D quadratic forms on the

pencil (6.8)-(6.9);

- Enforce a scale-sensitive (angular position, angular width) error model by a well-chosen algebraic transformation of this representation (6.13)-(6.14);
- Threshold this metric to find geometric inliers.

This method of matching keypoints with scales is very simple to use. In the following sections, we apply this method to evaluate keypoint detectors and present results on synthetic as well as real data.

6.3 Experiments

We now describe some illustrative experiments with the method on both synthetic data and a real image dataset.

6.3.1 Example

An example of matching SIFT points using only the angular mean constraint (6.13) and using both the angular mean and angular spread constraint (6.13)-(6.14) is shown in Figure 6.2. The combined constraint is considerably more selective.

6.3.2 Synthetic Data

We generate artificial scenes consisting of N 3D ellipsoids with random centres in the cube $[-1, 1]^3$, random scales distributed as s^{-2} in the interval $[0.005, 0.1]$, and random ellipticities with log-normal density of standard deviation 30%. These are viewed by two inwards-facing perspective cameras 4 units from the cube centre and 60° apart. The resulting image ellipses are perturbed in position and scale by Gaussian noise with standard deviation 33% of the ellipse radius. The ground truth epipolar geometry is used. Figure 6.3 (top left) shows an example of the image pairs generated and their epipolar geometry.

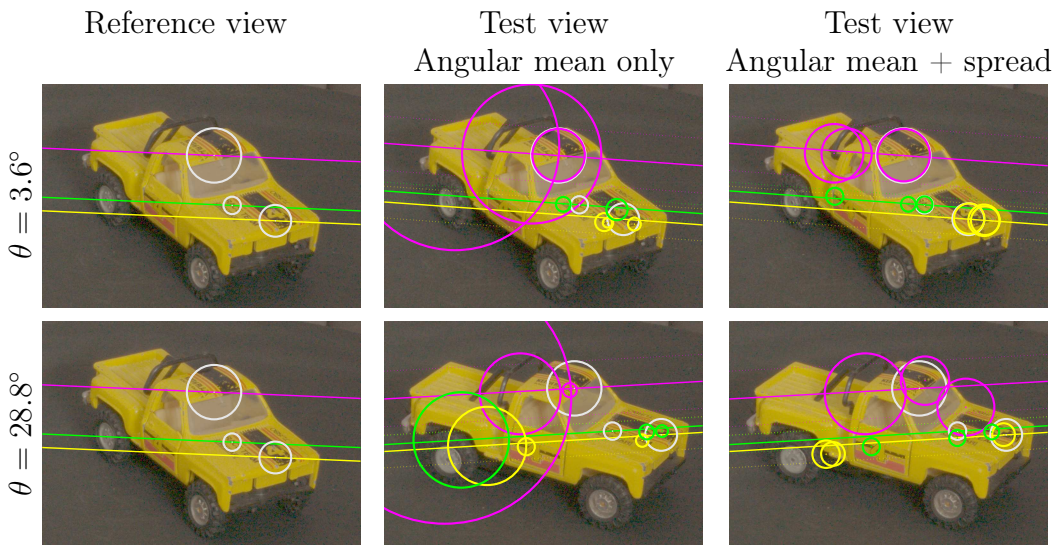


Figure 6.2: Example use of the method with SIFT keypoints on real images, for *Top:* 3.6° angular separation *Bottom:* 28.8° angular separation. Three keypoints (white circles) are selected in the left (reference) image. The corresponding centre and tangent epipolar lines are shown in both images. The three best matches to each point in the right (test) image are shown: under our angular mean ($d_{\bar{\theta}}$) decision rule *(middle)* and under our combined decision rule *(right)*. The true correspondences (marked manually) are shown as white circles. Note how the scale constraint (angular spread $d_{\delta\theta}$) is useful for pruning matches (right). Without this scale constraint, one often gets implausible matches (middle).

With these settings we find that for true correspondences, the errors underlying the $\bar{\theta}$ (6.13) and $\delta\theta$ (6.14) penalty terms are approximately jointly Gaussian (see Figure 6.3, bottom left and right), so that the penalties $d_{\bar{\theta}}, d_{\delta\theta}$ themselves have scaled 1 d.o.f. χ^2 distributions. In contrast, the distribution of errors for incorrect matches is much broader and is approximately uniform near the origin. This implies that a near-optimal inlier-outlier decision rule is to threshold the χ^2_2 variable $d_{\bar{\theta}}/\mu_{\bar{\theta}} + d_{\delta\theta}/\mu_{\delta\theta}$, where $\mu_{\bar{\theta}}, \mu_{\delta\theta}$ are the empirical means of the penalty functions (*i.e.* the variances of the underlying errors) for true matches. At a fixed percentage of false rejections, we find that using this rule reduces the number of false positives by a factor of around 2 to 2.5 relative to classical epipolar thresholding based on $d_{\bar{\theta}}$ alone. This gain holds across a wide range of feature densities N, rejection percentages, scene

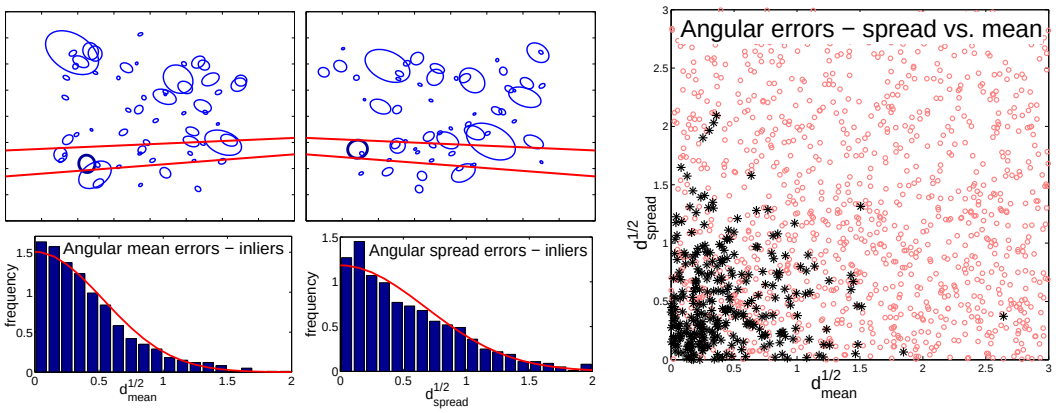


Figure 6.3: Experiments on synthetic data. Top left: left and right images of a scene containing random 3D ellipsoids, showing the epipolar lines tangent to a selected ellipse in the right image, and the corresponding lines in the left image almost tangent to the corresponding (noise perturbed) left ellipse. Bottom left: distributions of matching penalty values for correct but noise perturbed correspondences, for (left) the $\bar{\theta}$ (mean angle) penalty (6.13), (right) the $\delta\theta$ (angular spread) penalty (6.14). For each penalty d we histogram $\sqrt{d}$ (|linear error| rather than squared error) to show that the penalties behave roughly like χ^2_1 variables, *i.e.* the linear errors resemble half-Gaussians (red curves). Right: scatter plot of $\delta\theta$ penalty versus $\bar{\theta}$ penalty values over a large dataset. The black '$*$'s are correct matches and the red 'o's are incorrect ones. Again we plot linear errors $\sqrt{d}$. Clearly both the mean and the spread terms are useful for distinguishing inliers from outliers.

parameters, *etc.* It is increased by reduced uncertainty in, or broader distributions of, the ellipse scales, but we believe that our settings for these are representative of real detectors. For frontal camera motion (epipole at the image centre) the distribution of $\delta\theta$ values becomes somewhat broader and the gain is increased to around 4. Note that points with particularly large scales and ones that lie near the epipole are associated with broad sectors of epipolar lines, and therefore tend to match many other points under $d_{\bar{\theta}}$ alone. Adding $d_{\delta\theta}$ is particularly useful for eliminating these. On the other hand, there are typically many points with similar scales so $d_{\delta\theta}$ alone is not useful – it is only useful in combination with $d_{\bar{\theta}}$.

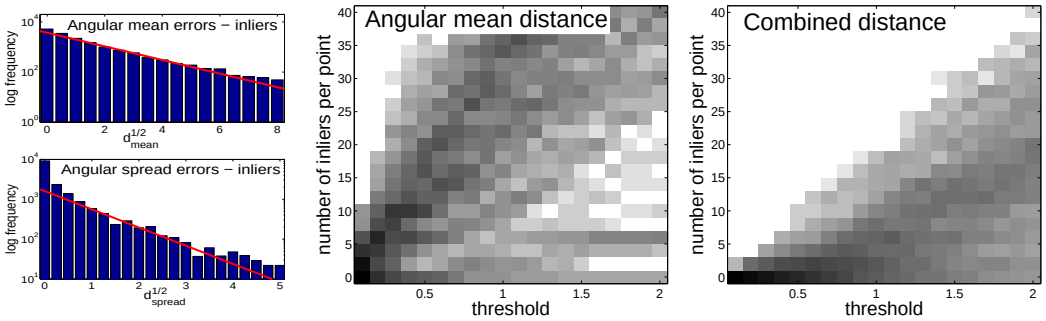


Figure 6.4: Experiments on real data. *Left*: For SIFT interest points, the distributions of $\sqrt{d_{\bar{\theta}}}$ and $\sqrt{d_{\delta\theta}}$ for corresponding features appear to be exponential with medians $m_{\bar{\theta}} \approx 1.0$ and $m_{\delta\theta} \approx 0.6$ (*i.e.* scale lengths 1.5 and 0.9). *Middle*: The histogram over an image pair of numbers of candidate matches satisfying the epipolar correspondence rule $\sqrt{d_{\bar{\theta}}}/m_{\bar{\theta}} < t$, for varying thresholds t. Darkness is proportional to log frequency. *Right*: The corresponding histogram for the combined decision rule $\sqrt{d_{\bar{\theta}}}/m_{\bar{\theta}} + \sqrt{d_{\delta\theta}}/m_{\delta\theta} < t$, which is approximately optimal for independent exponential variables against a uniform background of outliers. The combined rule is about 4 times more selective, producing many fewer incorrect correspondences.

6.3.3 Real data

We test the method using SIFT interest points on the real dataset described in Chapter 5 consisting of calibrated images of toy cars on a turn-table [Bendale et al., 2010b]. Similar to Chapter 5, we use the three image setup and obtain the ground truth by selecting correspondences that correspond both geometrically and by normalised cross-correlation patch matching. The resulting correspondences are far from perfect, but they suffice for an initial proof-of-concept test of the method on real data. Figure 6.4 (left) shows that the distributions of $\sqrt{d_{\bar{\theta}}}$ and $\sqrt{d_{\delta\theta}}$ for such 'inliers' are approximately exponential (Cauchy-like), not Gaussian. A corresponding scatter plot (not shown) demonstrates that the two error metrics again provide very complementary information for correspondence search. Figure 6.4 (middle) and (right) show that selecting possible correspondences by thresholding a weighted combination of the two metrics produces far fewer false matches than using epipolar line distances $d_{\bar{\theta}}$ alone. Similar conclusions are reached if the putative inliers for the tests are found using SIFT descriptor matching instead of 3-image

epipolar constraints.

6.4 Application to evaluation of keypoint detectors

The framework for evaluating keypoints that we developed in Chapter 5 can benefit from the enhanced multiscale epipolar matching constraints described here. We demonstrate this with the help of two examples here. First concerns the selection of correspondences between the reference and auxiliary images (reference–auxiliary pairs). The second example concerns the use of multiscale epipolar matching constraints based on ($d_{\bar{\theta}}$,$d_{\delta\theta}$) distances in place of conventional search for intersection of epipolar lines in the test image.

An example of selection of reference–auxiliary correspondences using our multiscale epipolar matching constraints is shown in Figure 6.5-6.6. For each reference image keypoint, we select a few (typically 4-5) putative matches amongst the auxiliary image keypoints based on some fixed threshold on the distances $d_{\bar{\theta}}$ and $d_{\delta\theta}$ (*c.f.* Figure 6.6). This is followed by normalised cross-correlation to select a single correspondence from the putative matches. The final reference–auxiliary matches correspond very well in scale as well as position.

Figure 6.7 shows the repeatability of SIFT and BTK-WT LS detectors using our multiscale epipolar matching method for a range of localisation error and scale mismatch thresholds. Each column shows the result of one setting of scale mismatch error $d_{\delta\theta}$ (also referred to as d_{spread}). From top to bottom, the allowed scale mismatch error reduces, thus making the matching constraint stricter. Within each graph, we have the result of three settings of localisation error threshold $d_{\bar{\theta}}$ (also referred to as d_{mean}). Within each graph, from top to bottom, the allowed localisation error reduces, making the matching constraint stricter. As might be expected, the graphs show that the repeatability reduces as a result of decreasing either the allowed localisation error or the allowed scale mismatch or both. Further, as we have seen before, SIFT has considerably better scale localisation as blobs are well localised in scale. BTK-WT LS shows good spatial localisation like SIFT, but picks up a mix of highly textured 2D features including corners, which

are generally less well-localised in scale.

The thresholds used in these experiments were manually chosen as small sensible values and the same values were used for both detectors. The exact values do not matter so much, the trend does. The aim was to highlight differences in performance of the two detectors used. These thresholds should ideally be determined empirically from known inlier statistics or based on a certain application, and will be the subject of our future work.

6.5 Conclusion

For both synthetic as well as real images, we find that, incorporating the additional scale constraint into the epipolar matching process cuts the number of false positive matches by a factor of 2–4 over a wide range of camera geometries and imaging conditions. Overall we conclude that for optimal results, evaluations of keypoint detectors should use both appearance information (similar to those presented in Chapter 5) and geometric information. Further integration of this framework with the evaluation framework described in Chapter 5 will be subject of our future work.

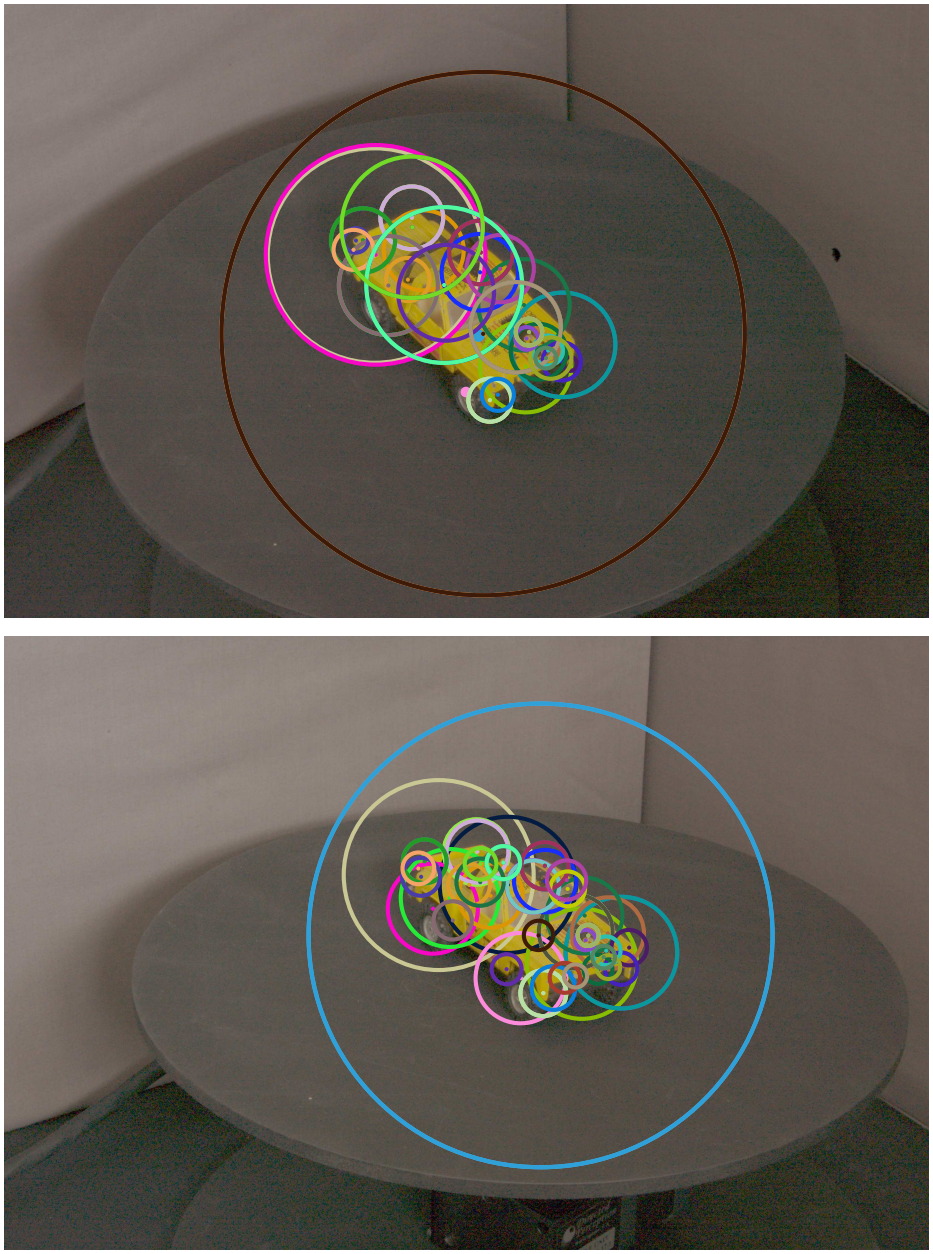


Figure 6.5: An example of selection of reference–auxiliary correspondences using our multiscale epipolar matching constraints, but using $d_{\bar{\theta}}$ only. Selected pairs are shown using the same colour in the *(bottom)* reference image and *(top)* auxiliary image. The final matches correspond in position but not in scale.

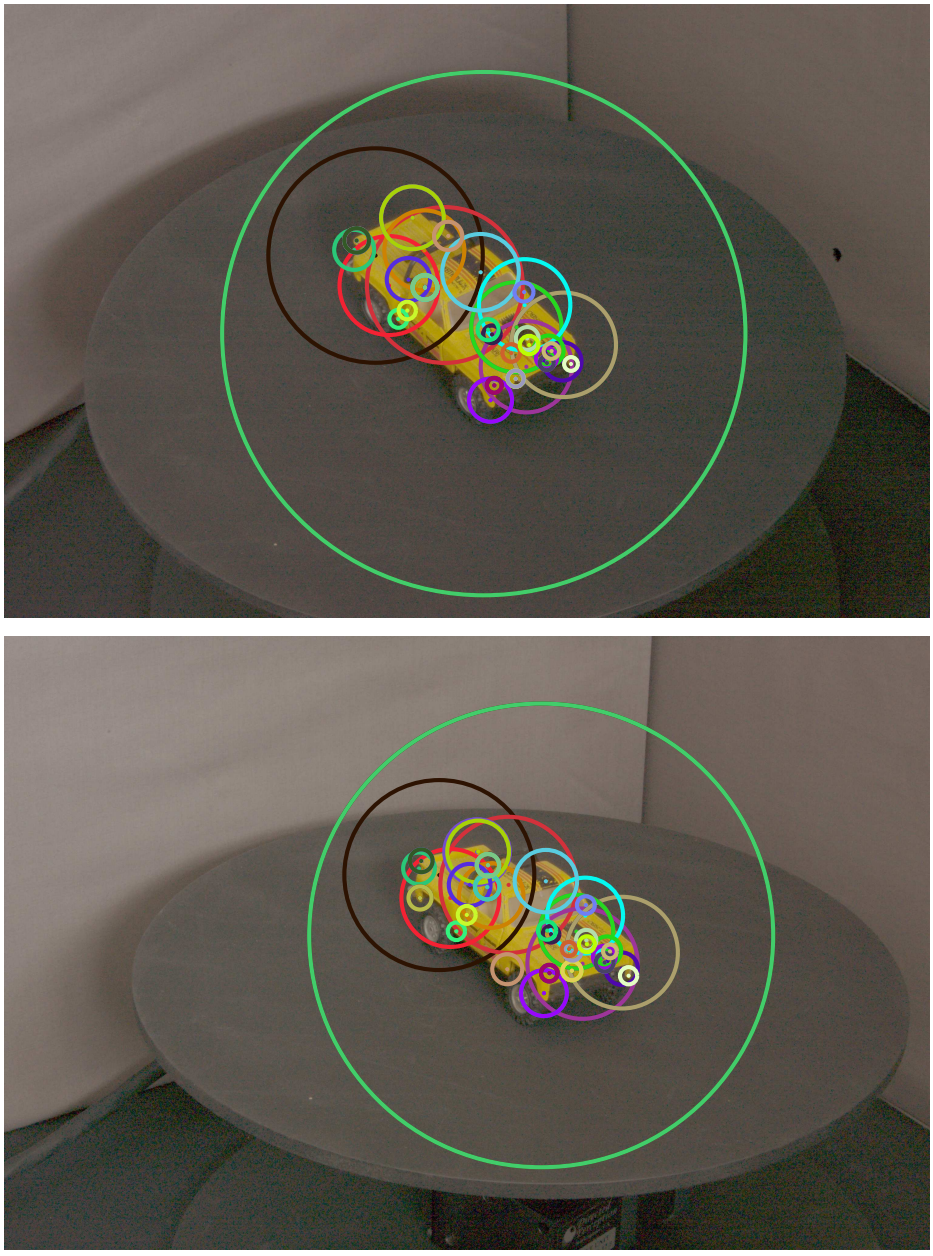


Figure 6.6: An example of selection of reference–auxiliary correspondences using our multiscale epipolar matching constraints on $d_{\bar{\theta}}$ and $d_{\delta\theta}$. Selected pairs are shown using the same colour in the *(bottom)* reference image and *(top)* auxiliary image. The final matches correspond in scale as well as position.

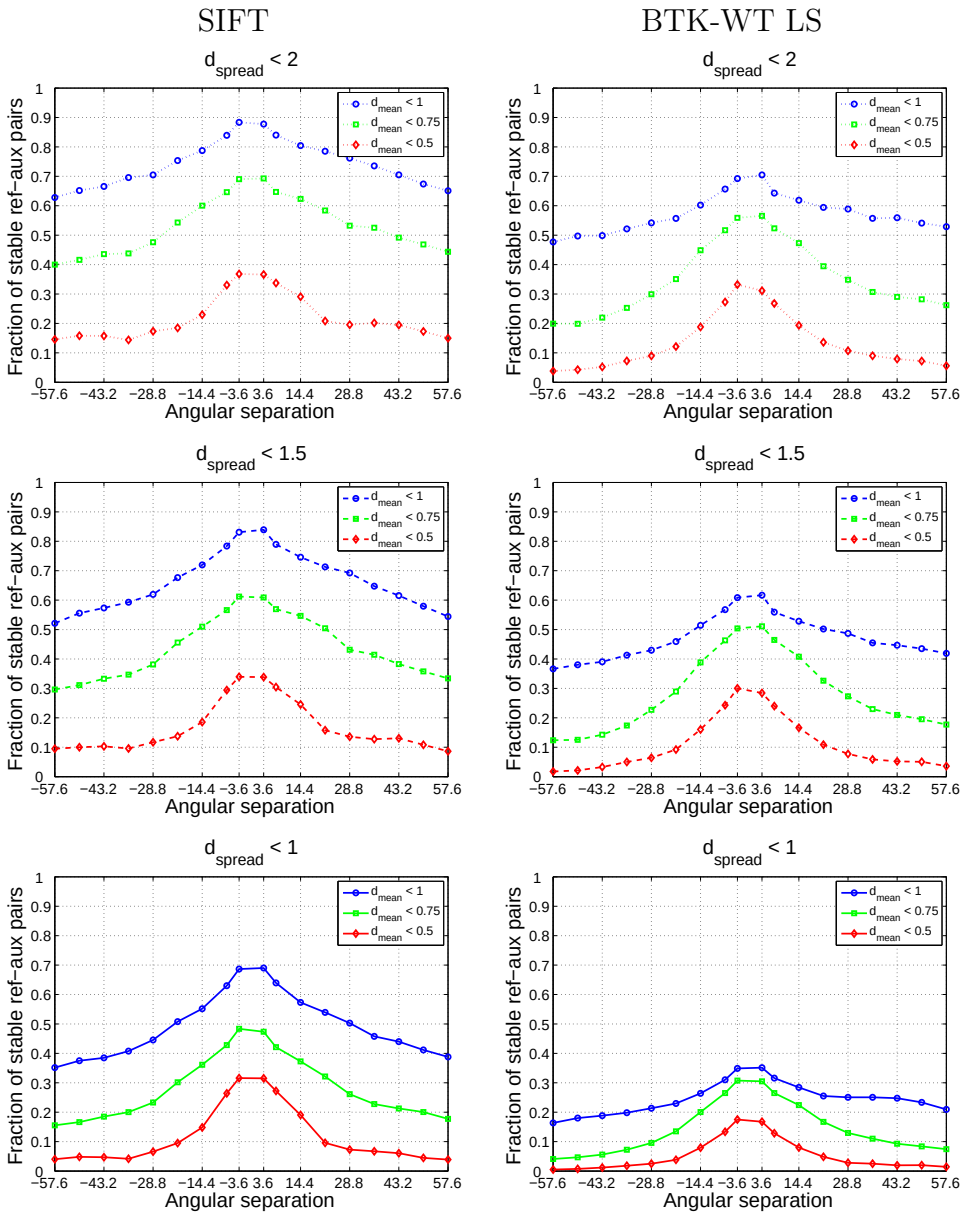


Figure 6.7: The repeatability of SIFT and BTK-WT LS detectors using our multiscale epipolar matching method. The reference–auxiliary pairs are selected using the method detailed in Figure 6.6. Using these, we plot the fraction of test image keypoints that satisfy the distance constraints using our multiscale epipolar matching constraints. The distances are used first to determine the match between reference image and test image and then between the auxiliary image and test image. The results are then combined, so that the results show the fraction of reference auxiliary pairs that satisfy a 3-way multiscale epipolar matching constraint.

Chapter 7

Conclusions and future work

This thesis was concerned with the development and evaluation of a wavelet-based keypoint detector and descriptor.

7.1 Conclusions

In *Chapter 3*, we began by developing the 4S-DTCWT scale-space framework. 4S-DTCWT provides both better-defined peaks and more consistent keypoint response amplitudes over scale. Subsequently, we examined in detail the types of keypoint responses at each level and strategies for keypoint localisation. We established that keypoint localisation in individual levels leads to better localisation in scale as well as better localisation in space, over a wider range of the scales available within the image. We investigated several methods for sub-pixel position and sub-level scale refinement in pyramidal scale-space, and least squares surface fitting with Gaussian weighting seems to work the best. All these changes led to a much improved feature detector, which we refer to as the BTK detector.

In *Chapter 4*, we adapted the Polar matching matrix descriptor for use within the 4S-DTCWT framework. We also formulated the computation of the descriptors and the matching scores in a simple and convenient matrix multiplication form. This leads to significant speedup of these operations, making the system more suitable for use in practical scenarios. We presented some new ways of evaluating keypoint methods.

In *Chapter 5*, we established that the performance of our keypoint detector and descriptor is now comparable to that of other methods. We presented the Cambridge toy cars dataset, a new 3D dataset, that we created to facilitate appearance-based evaluation of keypoint detectors and descriptors. This dataset provides the point to point mapping that is used as the ground truth for the experiments. Then we discussed the geometry of the setup and the test framework, followed by a quantitative evaluation of the repeatability of our keypoint detector and descriptor alongside other methods. We also presented quantitative results for some of the important configurations of the keypoint detector developed earlier in this thesis.

Finally, in *Chapter 6*, we presented an alternative multiscale geometry based method for evaluation of keypoint detectors which measures uncertainties in both scale and spatial positions and showed that it can be useful in evaluating the simultaneous scale and position stability of keypoints.

7.2 Future work

Where relevant, in the individual chapters, we discussed possibilities for future work. Here we present a summary.

- We have investigated a number of sub-pixel position and sub-level scale estimation methods, but we feel there is room for improvement. Specifically, we would like to explore non-isotropic Gaussian weighting for least squares method.

- The ability to interpolate 4S-DTCWT coefficients at any arbitrary spatial location gives us the flexibility to use fairly simple refinement methods for position. If this was true of scale as well, it would allow simpler methods for scale refinement and might lead to more stable scale estimates. Thus, inter-level interpolation of 4S-DTCWT coefficients (phase and magnitude) would be an interesting area for future work.

- We have primarily compared our detector, which is a more generic 2D feature detector that detects both corners and blobs, to the SIFT-DoG

detector, which focusses mainly on blob detection. It would be interesting to develop an equivalent blob detector within the 4S-DTCWT framework.

- We would like to make an open source C implementation of the detector and descriptor to facilitate researchers to use our detector and allow comparisons and further research in this area.

- Aspects other than repeatability can also be evaluated. These include the spatial spread of keypoints, the spread in detection scales, and the distinctiveness (uniqueness) of the detected keypoints.

- We would like to investigate the effect of transformations like pixelation, resampling, lens distortion, blurring, exposure *etc.* on keypoint localisation accuracy.

- A natural extension of our work would be to further integrate the multiscale epipolar constraints with descriptor based matching for the purpose of evaluation of keypoint detectors.

SIFT, Harris-Affine and Hessian-Affine methods were very much the best-in-class methods before we began our work and still are the best methods for generic keypoint matching. All these methods build upon earlier work on Gaussian scale spaces. However, the limited orientation selectivity of Gaussian scale spaces left something to be desired in terms of directional sensitivity. This project has opened doors to a new class of methods that seem to show promise of achieving similar or better performance without any additional computational cost. Previous research on parallelization of Gaussian pyramid computation is also directly applicable to the DTCWT pyramid computation, because at the heart of the two pyramids are repeated convolutions of an image with real, linearly separable filters. DTCWT scale space has attractive directional selectivity which, assuming everything else remains similar, should in theory lead to improved feature characterisation.

This project studied the DTCWT as a tool for feature characterisation (feature detection, description and matching). It investigated the potential

benefits of using the DTCWT as an alternative to the Gaussian scale space methods. DTCWT feature detection methods were relatively new to computer vision and early research [Fauqueur et al., 2006] in this area was a little disappointing (*c.f.* Figure 5.3), but further work in the area as described in this thesis has shown that DTCWT methods are capable of achieving near state-of-the-art performance. In the grand scheme of things, BTK detector is a useful addition to the battery of feature detectors, because it is reasonably stable and tends to detect somewhat different features in images.

Future work in this area might be concentrated on efficient implementations of the 4S-DTCWT pyramid. The idea of using phase of the DTCWT coefficients for incorporating tolerance to spatial shifts in the BTK descriptor is definitely worth exploring. Another area that is being actively researched at the moment involves making the descriptor tolerant to small shifts in keypoint position errors and small errors in keypoint scale [Nelson and Kingsbury, 2010] by pre-computing the derivatives of the descriptor and then estimating both the shift as well as the corrected matching score. All in all, DTCWT based feature characterisation methods have opened new and exciting avenues for research and we are confident they will contribute to the overall development of object recognition methods in the future.

Appendices

Appendix A

Epipolar geometry

For convenience, this appendix summarizes the basic concepts of two view epipolar geometry. For further details on multiple view geometry, see [Hartley and Zisserman, 2003]. We use the same notation here and base our description on [Hartley and Zisserman, 2003].

Consider a central pinhole camera located at the point $\mathbf{C}$ (camera centre) in a Euclidean 3-space (*c.f.* Figure A.1). Let the plane $Z = f$ be the image plane (or the focal plane). The *principal point*, $\mathbf{p} = (p_x, p_y)^\top$, is the point where the line perpendicular to the image plane passing through the camera centre intersects the image plane. This line is called the principal axis.

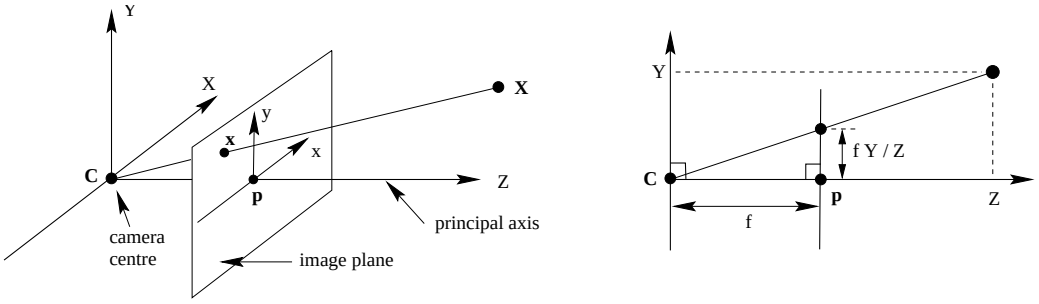


Figure A.1: Geometry of projection of a world point on an image plane via a central pinhole camera. $\mathbf{C}$ is the camera centre. $\mathbf{p}$ is the principal point. *Figure taken from [Hartley and Zisserman, 2003].*

A world point $\mathbf{X} = (X, Y, Z)^\top$ is projected onto the image plane at a point $\mathbf{x} = (x, y)$. The point $\mathbf{x}$ lies on the line joining the point $\mathbf{X}$ and the

camera centre **C**. For the moment, let **C** be the origin of the coordinate system. Then, $\mathbf{X} \mapsto \mathbf{x}$ implies

$$(X, Y, Z)^\top \mapsto (fX/Z + p_x, fY/Z + p_y)^\top = (x, y)^\top. \tag{A.1}$$

If we represent the world point **X** and the image point **x** in homogeneous coordinates as $\mathbf{X} = (X, Y, Z, 1)^\top$ and $\mathbf{x} = (x, y, f)$, then this mapping can be written as

$$\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \mapsto \begin{pmatrix} fX + Zp_x \\ fY + Zp_y \\ Z \end{pmatrix} = \begin{bmatrix} f & & p_x & 0 \\ & f & p_y & 0 \\ & & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \tag{A.2}$$

or, in matrix form, as

$$\mathbf{x} = \mathbf{K}\,[\mathbf{I} \mid \mathbf{0}]\,\mathbf{X} \tag{A.3}$$

with

$$\mathbf{K} = \begin{bmatrix} f & & p_x \\ & f & p_y \\ & & 1 \end{bmatrix}. \tag{A.4}$$

The 3×3 matrix **K** is called the *camera calibration matrix* (internal parameters).

Next, consider a world coordinate frame in which the origin of the Euclidean space is not at the camera centre **C**. The camera centre is related to the origin of this new coordinate system by a rotation **R** and a translation **t**, so that the world point $\mathbf{X} = \mathbf{R}\,\tilde{\mathbf{X}} + \mathbf{t}$, with $\mathbf{t} = -\mathbf{R}\,\tilde{\mathbf{C}}$. Substituting this in (A), we get

$$\mathbf{x} = \mathbf{K}\,\mathbf{R}\,[\mathbf{I} \mid -\tilde{\mathbf{C}}]\,\mathbf{X} \quad \text{or,} \tag{A.5}$$

$$\mathbf{x} = \mathbf{K}\,[\mathbf{R} \mid \mathbf{t}]\,\mathbf{X} \quad \text{or,} \tag{A.6}$$

$$\mathbf{x} = \mathbf{P}\,\mathbf{X}, \tag{A.7}$$

where $\mathbf{P} = \mathbf{K}\,[\mathbf{R} \,|\, \mathbf{t}]$ is the 3×4 homogeneous *camera projection matrix*. The rotation and translation matrices together comprise the external parameters.

If the imaging process is non-square, *i.e.* the image sensor has non-square pixels, then the focal length gets scaled unequally in the two directions. If s is the skew factor and m_x, m_y are the scale factors in x and y directions, then $\alpha_x = fm_x$ and $\alpha_y = fm_y$ represent the effective focal length in x and y directions respectively. The camera calibration matrix K may be rewritten to incorporate the skew as

$$\mathbf{K} \;=\; \begin{bmatrix} \alpha_x & s & x_0 \\ & \alpha_y & y_0 \\ & & 1 \end{bmatrix}. \tag{A.8}$$

Here, x_0, y_0 represent the principal point in world coordinates and α_y/α_x is the aspect ratio.

Figure A.2 shows the projection of a world point in two views. $\mathbf{C}$ and $\mathbf{C}'$ are the camera centres. The world point $\mathbf{X}$ is projected in two views as $\mathbf{x}$ and $\mathbf{x}'$. The intersections of the line joining the two camera centres with the two image planes are called the *epipoles* $\mathbf{e}$ and $\mathbf{e}'$ respectively. The plane containing the image points $\mathbf{x}$ and $\mathbf{x}'$, the world point $\mathbf{X}$, and the camera centres, is called the *epipolar plane* π. The intersection of the epipolar plane with the image plane is the *epipolar line*. All epipolar lines in an image intersect at the epipole.

$\mathbf{x}$ and $\mathbf{x}'$ are called correspondences because they are the projections of the same world point $\mathbf{X}$. The correspondent of $\mathbf{x}$ in the other view, $\mathbf{x}'$, is constrained to lie on the epipolar line $\mathbf{l}'$. The algebraic representation of the mapping $\mathbf{x} \mapsto \mathbf{l}'$ is given by a matrix called the *Fundamental matrix*. Thus,

$$\mathbf{l}' = \mathbf{F}\,\mathbf{x} \quad \text{and} \quad \mathbf{l} = \mathbf{F}^\top \mathbf{x}'. \tag{A.9}$$

is a correlation mapping that maps a point in one image to its corresponding epipolar line in the other image. The fundamental matrix is a rank 2 homogeneous matrix with 7 degrees of freedom. For all corresponding pairs

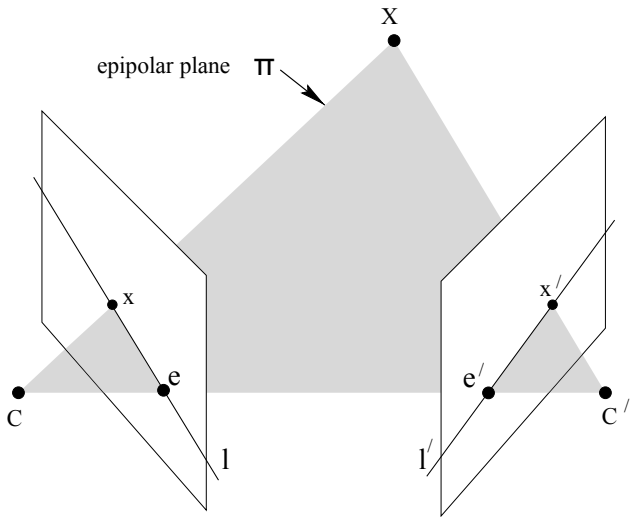


Figure A.2: Projection of a world point on two image planes via two cameras. *Figure derived from [Hartley and Zisserman, 2003].*

of points $(\mathbf{x}, \mathbf{x}')$, the following relation holds

$$\mathbf{x'}^{\top}\,\mathbf{F}\,\mathbf{x} = 0. \tag{A.10}$$

If the camera centre $\mathbf{C}$ is taken as the origin of the system, the projection matrix of the first camera may be expressed as $\mathbf{P} = \mathbf{K}\,[\mathbf{I} \mid \mathbf{0}]$. Further, if the rotation $(\mathbf{R})$ and translation $(\mathbf{t})$ of the second camera centre $\mathbf{C}'$ is known, then the projection matrix of the second camera is $\mathbf{P}' = \mathbf{K}'\,[\mathbf{R} \mid \mathbf{t}]$. The fundamental matrix may be expressed in terms of the two projection matrices as

$$\mathbf{F} = [\mathbf{e}']_{\times}\,\mathbf{P}'\,\mathbf{P}^{\dagger}, \tag{A.11}$$

where $[\mathbf{e}']_{\times}$ is the 3×3 skew cross product matrix of $\mathbf{e}'$ and $\mathbf{P}^{\dagger}$ is the pseudo-inverse of $\mathbf{P}$. More details may be found in [Xu and Zhang, 1996], [Luong and Faugeras, 1996] and [Hartley, 1997a].

Just like the bilinear relations Equations (A.10) enable us to derive the Fundamental matrix for two views, a set of trilinear relations allows us to come up with an entity called the *trifocal tensor*. The trifocal tensor is a

3×3×3 tensor that define the relations between corresponding points and corresponding lines in three views of a point.

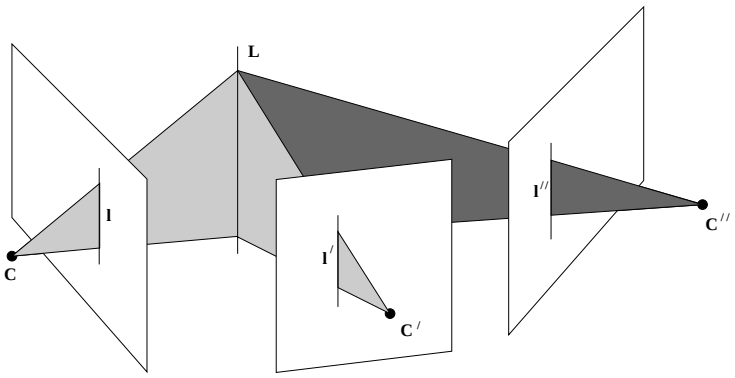


Figure A.3: Projection of a world point (or line) on three image planes via three cameras. *Figure taken from [Hartley and Zisserman, 2003].*

Figure A.3 shows a line **L** projected into three views. Unless the lines l, l' and l'' are the images of the line **L**, the planes containing the image lines will not intersect in a single line. This geometric constraint can also be extended to point incidence (the three back-projected rays from image points will intersect in a single point) and can be expressed algebraically in the form of a trifocal tensor.

The use of multiple view tensors is fairly involved and more details on the use of the trifocal tensor may be found in [Hartley and Zisserman, 2003](Chapter 14: The Trifocal Tensor and Appendix A1: Tensor Notation). The trifocal tensor was introduced and formalised in [Hartley, 1997b], but was known in various other forms through earlier work on trilinear relations (*c.f.* [Weng et al., 1988, Spetsakis and Aloimonos, 1991, Shashua, 1994, 1995]) and generalised to multiple images (Quadrifocal tensors for four views using quadrilinear relations and on to multiple view tensors) in [Triggs, 1995].

Appendix B

Epipolar constraints for multiscale matching

A paper published in the *British Machine Vision Conference, 2010* under the title

> "Epipolar constraints for multiscale matching"
> Bill Triggs, Pashmina Bendale

is included in the following ten pages.

The problem was identified while the authors were working on evaluation of keypoint detectors for [Bendale et al., 2010b]. Bill Triggs was responsible for developing and implementing the basic method and writing the theoretical part of the paper. Pashmina Bendale was responsible for implementing the test framework, running the tests, analysing the results and writing the experimental part of the paper.

Epipolar Constraints for Multiscale Matching

Bill Triggs
Bill.Triggs@imag.fr

Pashmina Bendale
pb397@cam.ac.uk

Laboratoire Jean Kuntzmann
BP 53, 38041 Grenoble Cedex 9
France
Signal Processing Laboratory
Department of Engineering
Cambridge CB2 1PZ, UK

Abstract

Many recent keypoint detectors associate a local scale (for multiscale detectors) or even a full affine frame (for affine-invariant detectors) to each detected keypoint. Although conventional epipolar constraints are a powerful tool for matching point-like features between pairs of images, they provide no constraint on their relative scales. We present an enhanced epipolar constraint that exploits these scales, thus providing more accurate correspondence search. The method encodes multiscale keypoints as image ellipses, invokes the classical Kruppa constraints that link corresponding ellipses, reduces these to constraints on 1-D quadratic forms on the pencil (1-D family) of epipolar lines, and enforces a scale-sensitive error model by a well-chosen algebraic transformation of the resulting homogeneous representation. The required projections onto the epipolar pencil are extracted from the Singular Value Decomposition of the Fundamental matrix. The final method is very simple to use. Illustrative tests yielded 2–4 fold reductions in false matches for both synthetic and real images. Matlab code is available.

1 Introduction

The conventional epipolar constraint is a powerful tool for matching keypoints (salient local features) between pairs of images, but in its standard form it treats the detected keypoints as point-like entities without intrinsic scales. In contrast, recent keypoint detectors (*c.f. e.g.* [8, 9] and their references) typically associate a scale (for multiscale detectors) or a full affine frame (for affine-invariant detectors) to each detection. In this paper we reformulate the epipolar constraint in a way that allows this additional scale information to be brought into play, thus providing a more selective overall matching process.

Specifically, we suppose that each detected keypoint can be described in terms of a circular or elliptical image region that characterizes the point's position, scale and shape, and that between pairs of images the ellipses of corresponding pairs of keypoints are in approximate correspondence, *i.e.* consistent with one another under the inter-image epipolar geometry. This will allow us to apply the classical Kruppa (conic correspondence) constraints[1]. We

This work was supported by a PhD grant from the Gates Cambridge Trust and by travel support from the European network PASCAL2. MATLAB code is available on http://ljk.imag.fr/membres/Bill.Triggs/src.

[1] Note that these constraints do not control the feature scales themselves, but rather the way that they must change (for true correspondences) as we move away from the epipole along the epipolar line.

will convert these to a simple angular 'epipolar pencil' representation that is convenient for matching, and construct an appropriate correspondence error metric in terms of this.

Before starting, we make several caveats. Firstly, note that our method is purely geometric. We treat ellipses as featureless geometric primitives, taking no account of the finer issue of establishing detailed pointwise correspondence between their underlying image content. Our approach is thus complementary to appearance-based correspondence methods such as multiscale or affine visual descriptors (*c.f. e.g.* [8, 10, 13] and their references). Both approaches provide useful constraints on correspondence and one should use whichever is most convenient for the problem in hand (and, when possible, combine them). In particular, if two keypoints correspond to the same salient region of a locally smooth surface in 3D, one would expect their ellipses to correspond under both our geometric constraints and appearance-based ones.

Secondly, throughout this paper we assume that the epipolar geometry (Fundamental matrix) is known. Any standard method can be used to estimate it [5]. We build an error model for uncertain multiscale points, not one for uncertain epipolar geometries.

Finally, note that even for fixed-scale keypoints, one can often model the uncertainties in their estimated positions with associated uncertainty ellipses. This is entirely different from the above use of ellipses to model associated image regions: although the resulting keypoint positions should still correspond modulo the uncertainties, there is no particular reason to expect their uncertainty ellipses to correspond because a keypoint may be localized well in one image and poorly in the other. Although it is not our focus here, our framework can handle this situation simply by omitting (13), the error term penalizing scale mismatch.

Prior Work. There has been a large amount of work on multiscale keypoint detectors and descriptors – *e.g.* [8, 9, 10, 12, 13], to give only a few examples. The use of epipolar geometry for inter-image keypoint matching and dense stereo is very well established, and the closely related 'Kruppa constraints' for correspondence between conics are also well known [3, 5, 6]. However there seems to be surprisingly little work dealing explicitly with integrating scales into epipolar geometry, perhaps because the authors interested in multiscale detection have tended to take descriptor-centric approaches not geometry-centric ones. The closest work that we are aware of is Forssén & Moe [4]. This uses elliptical blob features to estimate and apply epipolar geometry, but in a less unified way than the method given here.

2 Derivation of the Model

We now derive our model for multiscale keypoint correspondence via ellipses. The derivation takes some time but the final model is straightforward to use and is summarized at the end. We assume familiarity with the standard projective formulation of vision geometry (image projection, epipolar geometry, conics, *etc.*). For details see *e.g.* [5]. '$\sim$' denotes equality up to scale, and for a 3-vector $\mathbf{v}$, '$[\mathbf{v}]_\times$' denotes the corresponding 3×3 skew "cross product" matrix.

Epipolar Pencil Coordinates. Before starting on ellipses, we reexpress some basic facts about epipolar geometry in a useful form (*c.f. e.g.* [5] §18.4 'Kruppa Equations'). Let $\mathbf{P},\mathbf{P}'$ be the 3×4 perspective projection matrices of two images, called respectively 'left' and 'right' below. Let $\mathbf{F}$ be the corresponding fundamental matrix – $\mathbf{x}^\top\mathbf{F}\mathbf{x}' = 0$ is the epipolar constraint between left image point $\mathbf{x}$ and right image point $\mathbf{x}'$ – and let $\mathbf{e}$ and $\mathbf{e}'$ be the corresponding epipoles. From the Singular Value Decomposition (SVD) of $\mathbf{F}$, we can extract

the following representation

$$\mathbf{F} = \mathbf{U}\mathbf{S}\mathbf{V}^\top = \begin{pmatrix} -\mathbf{v} & \mathbf{u} & \mathbf{e} \end{pmatrix} \begin{pmatrix} \mu & & \\ & \nu & \\ & & 0 \end{pmatrix} \begin{pmatrix} \mathbf{u}' & \mathbf{v}' & \mathbf{e}' \end{pmatrix}^\top = \mathbf{u}\,\nu\,\mathbf{v}'^\top - \mathbf{v}\,\mu\,\mathbf{u}'^\top \tag{1}$$

Here, the 'twisted' naming in the U matrix of the SVD ensures that when viewed as points, the pairs $(\mathbf{u},\mathbf{u}')$ and $(\mathbf{v},\mathbf{v}')$ are in epipolar correspondence: $\mathbf{u}^\top\mathbf{F}\,\mathbf{u}' = 0 = \mathbf{v}^\top\mathbf{F}\,\mathbf{v}'$. Moreover, viewed as line-vectors, $\{\mathbf{u},\mathbf{v}\}$ and $\{\mathbf{u}',\mathbf{v}'\}$ respectively form bases for the pencils (1-D linear families) of epipolar lines in the left and right images, and $(\mathbf{u},\mathbf{u}')$ and $(\mathbf{v},\mathbf{v}')$ are corresponding pairs of epipolar lines: right point-vector $\mathbf{v}'$ lies on the right epipolar line $\mathbf{u}'$ (as $\mathbf{u}'^\top\mathbf{v}'$=0) and has left epipolar line $\mathbf{F}\,\mathbf{v}' \sim \mathbf{u}$, and vice versa. Finally, the left epipole $\mathbf{e}$ lies on the lines $\mathbf{u}$ and $\mathbf{v}$ and we can choose its sign so that $[\mathbf{e}]_\times = \mathbf{u}\,\mathbf{v}^\top - \mathbf{v}\,\mathbf{u}^\top$.

To make this more concrete, define the 2×3 "epipolar pencil projection" matrices[2]

$$\mathbf{B} \equiv \begin{pmatrix} \mathbf{u} & \mathbf{v} \end{pmatrix}^\top, \qquad \mathbf{B}' \equiv \begin{pmatrix} \mu\,\mathbf{u}' & \nu\,\mathbf{v}' \end{pmatrix}^\top \tag{2}$$

and note that

$$\mathbf{F} = \mathbf{B}^\top \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \mathbf{B}', \qquad [\mathbf{e}]_\times = \mathbf{u}\,\mathbf{v}^\top - \mathbf{v}\,\mathbf{u}^\top = \mathbf{B}^\top \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \mathbf{B}, \qquad \mathbf{B}\,\mathbf{B}^\top = \mathbf{I}_{2\times 2} \tag{3}$$

It follows that points $\mathbf{x},\mathbf{x}'$ are in epipolar correspondence, $\mathbf{x}^\top\mathbf{F}\,\mathbf{x}' = \mathbf{x}^\top\mathbf{B}^\top \left(\begin{smallmatrix} 0 & 1 \\ -1 & 0 \end{smallmatrix}\right) \mathbf{B}'\,\mathbf{x}' = 0$, if and only if the 2-vectors $\left(\begin{smallmatrix} \alpha \\ \beta \end{smallmatrix}\right) = \mathbf{B}\,\mathbf{x}$ and $\left(\begin{smallmatrix} \alpha' \\ \beta' \end{smallmatrix}\right) = \mathbf{B}'\,\mathbf{x}'$ are equal up to scale. We can view $(\alpha,\beta)^\top,(\alpha',\beta')^\top$ as homogeneous 2D coordinates for the 'projections' of $\mathbf{x},\mathbf{x}'$ onto the left epipolar pencil under $\mathbf{B},\mathbf{B}'$, with epipolar correspondence if and only if the projections agree up to scale. Moreover, $\mathbf{x}'$ has left epipolar line $\beta'\,\mathbf{u} - \alpha'\,\mathbf{v}$, and $\mathbf{x} = \alpha\,\mathbf{u} + \beta\,\mathbf{v} + \gamma\,\mathbf{e}$ lies on left epipolar line $\beta\,\mathbf{u} - \alpha\,\mathbf{v}$, so the corresponding epipolar lines have the dual coordinates $(\beta,-\alpha)$ and $(\beta',-\alpha')$. Thus, the projections $\mathbf{B},\mathbf{B}'$ allow us to reduce image-based epipolar geometry calculations to 1D epipolar pencil ones. We will apply this to conics below. The appendix shows how to extend this to spherical images and signed epipolar geometry.

Conic Correspondence on the Epipolar Pencil. The geometry of projections and epipolar constraints for quadrics is well known [5] – see fig. 1. Let Q be a 3D quadric – here typically a (possibly planar) ellipsoid. Its images in the two cameras are 2D conics q,q' representing the envelopes of the image rays tangent to Q. Exactly two epipolar planes are tangent to Q in 3D, and in each image these generate a pair of corresponding epipolar lines that are tangent to the conics q,q'. Conversely, given any two image conics q,q' with corresponding epipolar lines, it turns out that there is a 3D quadric Q (in fact a 1 parameter family of them) whose images are q and q'. The epipolar constraint between conics is thus the requirement that the two epipolar lines tangent to q are in epipolar correspondence with the two epipolar lines tangent to q'.

In formulae, if we represent Q in dual (hyperplane) form by 4×4 symmetric matrix $\mathbf{Q}$ and q,q' in dual (line) form by 3×3 symmetric matrices $\mathbf{q},\mathbf{q}'$, the image projections are simply

[2] Well-normalized image coordinates should be used to evaluate the $\mathbf{B}$'s: $\mathbf{F} \to \mathbf{K}_1^{-\top}\mathbf{F}\,\mathbf{K}_2^{-1}$ on input to the SVD and $\mathbf{B}_1 \to \mathbf{B}_1\,\mathbf{K}_1$, $\mathbf{B}_2 \to \mathbf{B}_2\,\mathbf{K}_2$ on output, where $\mathbf{K}_i \sim \begin{pmatrix} 1/f_x & & -p_x/f_x \\ & 1/f_y & -p_y/f_y \\ & & 1 \end{pmatrix}$ are nominal calibration matrices with "focal lengths" / normalization scales f_x,f_y (in pixels) and "principal point" / image centre $(p_x,p_y)^\top$. The exact values used are not critical, but if unnormalized pixel coordinates are used the resulting epipolar pencil coordinates tend to be very distorted, leading to unintuitive behaviour in the below angular error estimates.

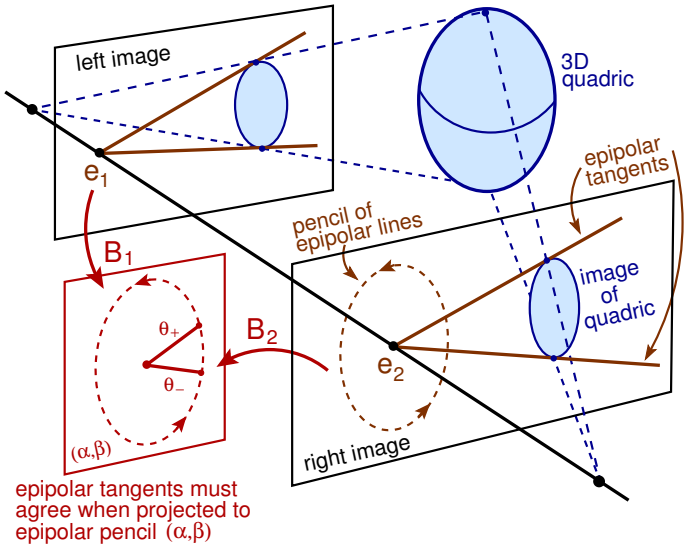


Figure 1: Projection of a 3D quadric to two image conics, and then projection via $\mathbf{B},\mathbf{B}'$ of the pair of epipolar lines tangent to each conic (or equivalently, of the quadratic form defined by the conic) to the epipolar pencil. The reduced Kruppa constraints state that the two projections agree.

$\mathbf{q} \sim \mathbf{P}\mathbf{Q}\mathbf{P}^\top$ and $\mathbf{q}' \sim \mathbf{P}'\mathbf{Q}\mathbf{P}'^\top$ [11]. More explicitly, the dual form of an image ellipse with 2D centre $\boldsymbol{c}$ and 2D covariance matrix $\boldsymbol{V}$ is

$$\mathbf{q} = \begin{pmatrix} \boldsymbol{c}\boldsymbol{c}^\top - \boldsymbol{V} & \boldsymbol{c} \\ \boldsymbol{c}^\top & 1 \end{pmatrix} \tag{4}$$

so an image line $\boldsymbol{n}^\top \binom{x}{y} - d = 0$ is tangent to q if and only if $\begin{pmatrix} \boldsymbol{n}^\top & -d \end{pmatrix} \mathbf{q} \binom{\boldsymbol{n}}{-d} = (\boldsymbol{n}^\top \boldsymbol{c} - d)^2 - \boldsymbol{n}'^\top \boldsymbol{V}\boldsymbol{n} = 0$. For a circle, $\boldsymbol{V} = r^2\boldsymbol{I}$, the tangency equation becomes $(\boldsymbol{n}^\top \boldsymbol{c} - d)^2 = r^2$, *i.e.* any line that passes at distance r from the centre $\boldsymbol{c}$ is tangent to the circle. Analogous forms hold for the 3D quadric $\mathbf{Q}$ and its tangent planes, with 3D $\boldsymbol{c},\boldsymbol{V},\boldsymbol{n}$.

In this framework, conic correspondence turns out to be encapsulated by the 'Kruppa constraints'[3] [3, 5]

$$[\mathbf{e}]_\times \, \mathbf{q} \, [\mathbf{e}]_\times^\top \;\sim\; \mathbf{F}\mathbf{q}'\mathbf{F}^\top \tag{5}$$

Each side of this equation is a 3×3 symmetric rank 2 matrix of the form $\mathbf{l}_1\mathbf{l}_2^\top + \mathbf{l}_2\mathbf{l}_1^\top$ where $\mathbf{l}_1,\mathbf{l}_2$ are the left epipolar lines tangent q or q', as appropriate. Applying (3) and cancelling extraneous factors of $\mathbf{B}^\top \left(\begin{smallmatrix} 0 & 1 \\ -1 & 0 \end{smallmatrix}\right)$ gives the 2×2 symmetric 'Reduced Kruppa Constraints'[4] [5]

$$\mathbf{B}\mathbf{q}\mathbf{B}^\top \sim \mathbf{B}'\mathbf{q}'\mathbf{B}'^\top, \qquad \text{or in coordinate form} \qquad \begin{pmatrix} q_{vv} \\ q_{uv} \\ q_{uu} \end{pmatrix} \sim \begin{pmatrix} v^2 q'_{vv} \\ \mu v \, q'_{uv} \\ \mu^2 q'_{uu} \end{pmatrix} \tag{6}$$

[3] An alternative form is that conics correspond iff the symmetric part of $[\mathbf{e}]_\times \, \mathbf{q}\mathbf{F}\mathbf{q}'\mathbf{F}^\top$ vanishes. Under correspondence this matrix is $\sim [\mathbf{e}]_\times$. *C.f.* a homography $\mathbf{H}$ is consistent with $\mathbf{F}$ iff $\mathbf{F}\mathbf{H}$ is skew (and $\sim [e]_\times$).

[4] Kruppa constraints were originally developed for camera autocalibration [3, 5]. In this context, $\mathbf{q},\mathbf{q}'$ are chosen to be the "images of the dual absolute quadric" $\mathbf{q} = \mathbf{K}\mathbf{K}^\top$, $\mathbf{q}' = \mathbf{K}'\mathbf{K}'^\top$, where $\mathbf{K},\mathbf{K}'$ are the camera internal parameter matrices. Their reduced quadratic (6) determines the metric circle structure ("circular points") of (α,β) – *i.e.* which pairs of epipolar lines correspond to orthogonal pairs of 3D epipolar planes.

where the coordinates are $q_{uv} \equiv \mathbf{u}^\top \mathbf{q}\, \mathbf{v}$, $q'_{uu} \equiv \mathbf{u}'^\top \mathbf{q}'\, \mathbf{u}'$, *etc*. These equations provide two independent algebraic constraints representing the two epipolar tangencies. Below we will use $(a,b,c)^\top$ to denote either the left or right hand side of (6), as needed.

Evaluating the dual conic $\mathbf{q}$ on the epipolar line vector corresponding to 'point' $(\alpha,\beta)^\top$ induces a corresponding reduced quadratic form on the pencil

$$\begin{pmatrix}\beta & -\alpha\end{pmatrix} \mathbf{B}\,\mathbf{q}\,\mathbf{B}^\top \begin{pmatrix}\beta \\ -\alpha\end{pmatrix} \;=\; a\,\alpha^2 - 2\,b\,\alpha\beta + c\beta^2 \tag{7}$$

and similarly for $\mathbf{q}'$ with $\mathbf{B}'$. For example, inserting (4) into (7) gives $(\alpha_c\,\beta - \beta_c\,\alpha)^2 - (\beta\ -\alpha)\,\tilde{\boldsymbol{V}} \begin{pmatrix}\beta \\ -\alpha\end{pmatrix}$ where $\begin{pmatrix}\alpha_c \\ \beta_c\end{pmatrix} \equiv \mathbf{B}\begin{pmatrix}c \\ 1\end{pmatrix}$ and $\tilde{\boldsymbol{V}} \equiv \mathbf{B}\begin{pmatrix}\boldsymbol{V} & \mathbf{0} \\ \mathbf{0} & 0\end{pmatrix}\mathbf{B}^\top$. If $\boldsymbol{V} \to \boldsymbol{0}$ this has a double root at $(\alpha_c,\beta_c)^\top$ as expected.

Algebraic Weakness of Kruppa Form. To gain further intuition, we briefly switch to angular coordinates $(\alpha,\beta)^\top \sim (\cos\theta,\sin\theta)^\top$, letting the roots of the reduced quadratic be[5] $\theta_\pm = \bar{\theta} \pm \delta\theta$. Then up to scale, (7) becomes

$$\begin{aligned}\sin(\theta-\theta_+)\sin(\theta-\theta_-) \;&=\; (\sin\theta\cos\theta_+ - \cos\theta\sin\theta_+)(\sin\theta\cos\theta_- - \cos\theta\sin\theta_-) &(8)\\ &=\; \big(\cos 2\delta\theta - \cos 2(\theta-\bar{\theta})\big)/2 &(9)\end{aligned}$$

where the various forms follow from standard trigonometric identities, and

$$\begin{pmatrix}a\\b\\c\end{pmatrix} \sim \begin{pmatrix}\sin\theta_+\sin\theta_- \\ (\cos\theta_+\sin\theta_- + \sin\theta_+\cos\theta_-)/2 \\ \cos\theta_+\cos\theta_-\end{pmatrix} = \frac{1}{2}\begin{pmatrix}\cos 2\delta\theta - \cos 2\bar{\theta} \\ \sin 2\bar{\theta} \\ \cos 2\delta\theta + \cos 2\bar{\theta}\end{pmatrix} \tag{10}$$

$$\begin{pmatrix}p\\q\\r\end{pmatrix} \equiv \begin{pmatrix}c-a\\2b\\c+a\end{pmatrix} \sim \begin{pmatrix}\cos 2\bar{\theta}\\ \sin 2\bar{\theta}\\ \cos 2\delta\theta\end{pmatrix} \tag{11}$$

We would like to use reduced Kruppa vectors $(a,b,c)^\top$ or $(p,q,r)^\top$ to quantify the extent to which two ellipses are in epipolar correspondence. The $(\cos 2\bar{\theta},\sin 2\bar{\theta})$ components of (11) should provide good control over errors in the mean epipolar line direction $\bar{\theta}$. Unfortunately, the $\cos 2\delta\theta$ term provides much less control over $\delta\theta$. In particular, for ellipses that are small relative to their distance to the epipole, $\delta\theta$ is small and $\cos^2 2\delta\theta \approx 1 - 4\delta\theta^2$ provides only a weak second order constraint on it: the Kruppa constraints are algebraically complete but unsuitable as error models because they provide too little control over the sizes of the small ellipses that make up most of the keypoint population. One way around this would be to introduce a fourth coordinate $s \equiv \sqrt{p^2+q^2-r^2} \sim \sin 2\delta\theta$ into (11), to give better control of $\delta\theta$. We will embed an analogous transformation in our error metric. Henceforth we suppose that $(p,q,r)^\top$ has been rescaled to make $p^2+q^2=1$, so that (11) becomes an equality.

Epipolar Pencil Error Model. We need to design an error weighting that reflects our expectations regarding uncertainties in keypoint positions and scales. A great many models are possible. Here we develop just one as an example, based on the assumption that both the position uncertainty and the scale uncertainty of a typical keypoint are proportional to

[5] To ensure consistent signs we take $\theta_- \le \theta_+ < \theta_- + \pi$, *i.e.* $0 \le \delta\theta < \frac{\pi}{2}$. Also, we assume that the quadratics (7) have real roots ($b^2 \ge 4ac$, or $r^2 \le p^2+q^2 = 1$ below), *i.e.* that neither image ellipse contains (surrounds) its epipole. This is true for most keypoints. If not, we can either discard the point, or note that (5), (6) still provide valid constraints on the match – the geometric interpretation is now that the circular points implied on the line $(\alpha,\beta)^\top$ by the two quadratics must agree – and, *e.g.*, use $r-1$ in place of $1-r$ in (12), (13) below, noting that for a match of this kind, *both* quadratics must have $r > 1$.

its scale and that deviations are approximately Gaussian. To concretize this algebraically on the epipolar pencil we will use $\sigma \equiv \sqrt{(1-r)/2} = \sin\delta\theta$ as a surrogate for the ellipse scale, as this scales linearly with $\delta\theta$ at small angles and increases smoothly to 1 at very large ones $\delta\theta \approx \frac{\pi}{2}$ or $\theta_+ - \theta_- \approx \pi$. (Moreover, $-\sigma^2 = (r-1)/2$ is the minimum value of (8)). Assuming independent Gaussian errors of (small) standard deviation $\delta\theta$ in $\bar{\theta}$, the standard (two-sample z-test) statistic for deviations of the means is $(\bar{\theta}-\bar{\theta}')^2/(\delta\theta^2+\delta\theta'^2)$. We will algebraize this as

$$d_{\bar{\theta}} \equiv \frac{\sin^2 2(\bar{\theta}-\bar{\theta}')}{\sin^2\delta\theta+\sin^2\delta\theta'} = \frac{(pq'-qp')^2}{\sigma^2+\sigma'^2} = \frac{(pq'-qp')^2}{1-(r+r')/2} \tag{12}$$

Various penalties (log-normal, *etc.*) are possible for the errors in $\delta\theta$, but one viable statistic is $\left(\delta\theta/\delta\theta'\right)^k + \left(\delta\theta'/\delta\theta\right)^k - 2$ for some k. This is symmetric, scale invariant, zero at equality and grows as $O(\delta\theta^{-k})$ when either $\delta\theta$ shrinks to zero with the other held constant. We will algebraize this as

$$d_{\delta\theta} \equiv \left(\frac{\sin\delta\theta}{\sin\delta\theta'}\right)^k + \left(\frac{\sin\delta\theta'}{\sin\delta\theta}\right)^k - 2 = \left(\frac{\sigma}{\sigma'}\right)^k + \left(\frac{\sigma'}{\sigma}\right)^k - 2 = \left(\frac{1-r}{1-r'}\right)^{k/2} + \left(\frac{1-r'}{1-r}\right)^{k/2} - 2 \tag{13}$$

Our final error model will be a weighted[6] sum of (12) and (13), and for simplicity we will use $k=2$ below. Although we will not consider them here, other more heuristic error models could also be used, for example weighting squared distances between $(p,q,r,s)^\top$ vectors with a suitable function such as $\frac{1}{\sigma^2+\sigma'^2}$, $\frac{1}{\sigma\sigma'}$ or $\frac{1}{\sigma^2}+\frac{1}{\sigma'^2}$.

Summary of Method. The final method is very straightforward. For each ellipse, the matrix $\mathbf{q}$ or $\mathbf{q}'$ (4) is projected using $\mathbf{B}$ or $\mathbf{B}'$ (from the SVD of $\mathbf{F}$) to obtain $(a,b,c)^\top$ (6), normalized to give $(p,q,r)^\top$ (11), and – see appendix (14) – optionally unwrapped to signed form. 'Distances' between these vectors are then computed using a weighted sum of (12) and (13) and used to decide whether pairs of ellipses might correspond.

Further Points. Here we opted for algebraic representations for simplicity, but it would also have been possible to explicitly recover and use $\bar{\theta}, \delta\theta$. In particular, if an efficient data structure for inlier search is required, one could represent search intervals as rectangles in $(\bar{\theta}, \delta\theta)$ coordinates and use some data structure such as a box tree that allows efficient search for all of the left image rectangles containing an observed pair $(\bar{\theta}', \delta\theta')$ from the right image. This would require cutting the $\bar{\theta}$ circle to a flat interval $[0, 2\pi]$ (so some points would generate two rectangles), and using the given $(\bar{\theta}, \delta\theta)$ value, the acceptance thresholds on (12)-(13) and, for (12), a bound on r': $r' \in [-1, 1]$, to derive search bounds on $(\bar{\theta}', \delta\theta')$.

The main limitation of the epipolar pencil representation is that it suppresses all information about positions along epipolar lines. If such information is useful[7], it must be applied separately from the epipolar pencil method. The above derivations also assume that $\mathbf{F}, \mathbf{B}, \mathbf{B}'$ are known exactly and we currently make no effort to incorporate uncertainties in these into the computations. Although the resulting effective geometric search regions (wedge-shaped ones starting at the epipoles) are actually statistically more correct and simpler to use than the common parallel strips along epipolar lines, in practice it is wise to allow a few additional pixels of slack to account for uncertainties in $\mathbf{F}$, especially for keypoints near the epipoles.

[6] *E.g.*, with the above weightings and for small $\delta\theta$, a factor-of-2 error in $\delta\theta$ has about the same penalty as a position error of $2\delta\theta$.

[7] For example, if the scene is bounded by known 3D half spaces or by the plane at infinity, their homographies can be used to limit the search for a given point's correspondent along its epipolar line.

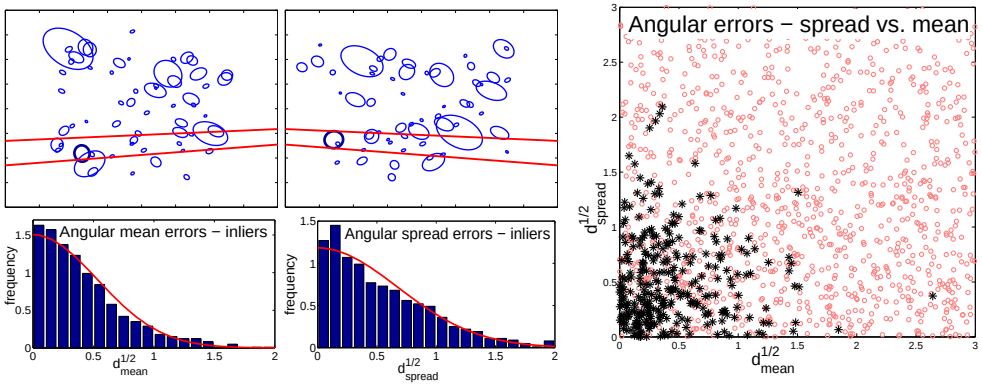


Figure 2: Experiments on synthetic data. Top left: left and right images of a scene containing random 3D ellipsoids, showing the epipolar lines tangent to a selected ellipse in the right image, and the corresponding lines in the left image almost tangent to the corresponding (noise perturbed) left ellipse. Bottom left: distributions of matching penalty values for correct but noise perturbed correspondences, for (left) the $\bar{\theta}$ (mean angle) penalty (12), (right) the $\delta\theta$ (angular spread) penalty (13). For each penalty d we histogram $\sqrt{d}$ (|linear error| rather than squared error) to show that the penalties behave roughly like χ^2_1 variables, *i.e.* the linear errors resemble half-Gaussians (red curves). Right: scatter plot of $\delta\theta$ penalty versus $\bar{\theta}$ penalty values over a large dataset. The black '*'s are correct matches and the red 'o's are incorrect ones. Again we plot linear errors $\sqrt{d}$. Clearly both the mean and the spread terms are useful for distinguishing inliers from outliers.

3 Experiments

We now describe some illustrative experiments with the method[8] on both synthetic data and a real image dataset. A more detailed study will be published later.

Synthetic Data. We generate artificial scenes consisting of N 3D ellipsoids with random centres in the cube $[-1,1]^3$, random scales distributed as s^{-2} in the interval $[0.005, 0.1]$, and random ellipticities with log-normal density of standard deviation 30%. These are viewed by two inwards-facing perspective cameras 4 units from the cube centre and 60° apart. The resulting image ellipses are perturbed in position and scale by Gaussian noise with standard deviation 33% of the ellipse radius. The ground truth epipolar geometry is used. Fig. 2 (top left) shows an example of the image pairs generated and their epipolar geometry.

With these settings we find that for true correspondences, the errors underlying the $\bar{\theta}$ (12) and $\delta\theta$ (13) penalty terms are approximately jointly Gaussian (see fig. 2, bottom left and right), so that the penalties $d_{\bar{\theta}}, d_{\delta\theta}$ themselves have scaled 1 d.o.f. χ^2 distributions. In contrast, the distribution of errors for incorrect matches is much broader and is approximately uniform near the origin. This implies that a near-optimal inlier-outlier decision rule is to threshold the χ^2_2 variable $d_{\bar{\theta}}/\mu_{\bar{\theta}} + d_{\delta\theta}/\mu_{\delta\theta}$, where $\mu_{\bar{\theta}}, \mu_{\delta\theta}$ are the empirical means of the penalty functions (*i.e.* the variances of the underlying errors) for true matches. At a fixed percentage of false rejections, we find that using this rule reduces the number of false positives by a factor of around 2 to 2.5 relative to classical epipolar thresholding based on $d_{\bar{\theta}}$ alone. This gain holds across a wide range of feature densities N, rejection percentages,

[8] A Matlab implementation is available from http://ljk.imag.fr/membres/Bill.Triggs/src.

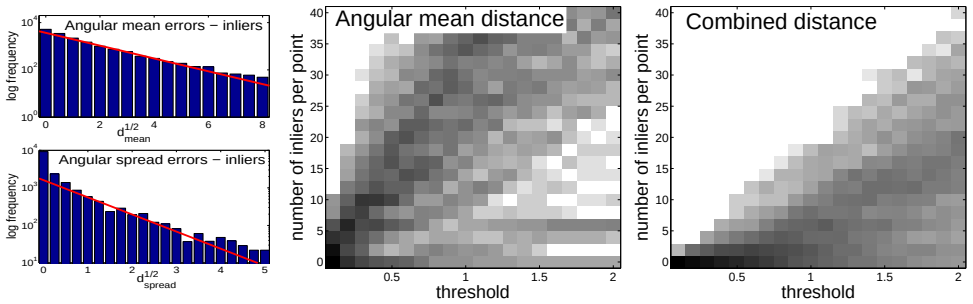


Figure 3: Experiments on real data. (Left) For SIFT interest points, the distributions of $\sqrt{d_{\bar{\theta}}}$ and $\sqrt{d_{\delta\theta}}$ for corresponding features appear to be exponential with medians $m_{\bar{\theta}} \approx 1.0$ and $m_{\delta\theta} \approx 0.6$ (*i.e.* scale lengths 1.5 and 0.9). (Middle) The histogram over an image pair of numbers of candidate matches satisfing the epipolar correspondence rule $\sqrt{d_{\bar{\theta}}}/m_{\bar{\theta}} < t$, for varying thresholds t. Darkness is proportional to log frequency. (Right) The corresponding histogram for the combined decision rule $\sqrt{d_{\bar{\theta}}}/m_{\bar{\theta}} + \sqrt{d_{\delta\theta}}/m_{\delta\theta} < t$, which is approximately optimal for independent exponential variables against a uniform background of outliers. The combined rule is about 4 times more selective, producing many fewer incorrect correspondences.

scene parameters, *etc*. It is increased by reduced uncertainty in, or broader distributions of, the ellipse scales, but we believe that our settings for these are representative of real detectors. For frontal camera motion (epipole at the image centre) the distribution of $\delta\theta$ values becomes somewhat broader and the gain is increased to around 4. Note that points with particularly large scales and ones that lie near the epipole are associated with broad sectors of epipolar lines, and therefore tend to match many other points under $d_{\bar{\theta}}$ alone. Adding $d_{\delta\theta}$ is particularly useful for eliminating these. On the other hand, there are typically many points with similar scales so $d_{\delta\theta}$ alone is not useful – it is only useful in combination with $d_{\bar{\theta}}$.

Real data. We also tested the method using SIFT interest points on a real dataset consisting of calibrated images of toy cars on a turntable[9] [1]. As a surrogate for ground-truth correspondences, we used conventional epipolar constraints over a triangle of images – 'Reference', 'Auxilliary' and 'Test' – selecting point pairs that corresponded both geometrically and by least squares patch matching in Reference and Auxilliary, and accepting any point within a generous region around the intersection of their two epipolar lines in Test as an inlier for the purposes of the evaluation. The resulting correspondences are far from perfect, but they suffice for an initial proof-of-concept test of the method on real data. Fig. 3 (left) shows that the distributions of $\sqrt{d_{\bar{\theta}}}$ and $\sqrt{d_{\delta\theta}}$ for such 'inliers' are approximately exponential (Cauchy-like), not Gaussian. A corresponding scatter plot (not shown) demonstrates that the two error metrics again provide very complementary information for correspondence search. Fig. 3 (middle) and (right) show that selecting possible correspondences by thresholding a weighted combination of the two metrics produces far fewer false matches than using epipolar line distances $d_{\bar{\theta}}$ alone. Similar conclusions are reached if the putative inliers for the tests are found using SIFT descriptor matching instead of 3-image epipolar constraints.

[9] http://www-sigproc.eng.cam.ac.uk/imu

4 Summary and Conclusions

We have introduced a framework for epipolar correspondence search that provides tighter constraints for matching multiscale keypoints by constraining the features' relative scales as well as their positions. The method works by representing uncertain keypoints as image ellipses, using 2×3 projection matrices $\mathbf{B},\mathbf{B}'$ extracted from the fundamental matrix $\mathbf{F}$ to project these onto the epipolar pencil – *i.e.* w.r.t. $(\alpha,\beta)^\top$ coordinates on the 1-D family of left epipolar lines – and formulating the quality of the match in terms of an algebraic error model in these coordinates. The method is elegant, simple to use (probably simpler than the traditional image search along epipolar strips) and gives a substantial reduction in false correspondences – typically a factor of 2–4 in our synthetic and real experiments.

5 Appendix: Spherical Images and Unwrapping

This appendix extends the above epipolar pencil constructions to the 'spherical' (also called 'oriented' or 'signed') approach to projective vision in which image points are identified with points on the camera's viewing sphere (the sphere of incoming 3D visual rays at its centre) [2, 7]. In this framework, epipolar lines correspond to *half* great circles joining the epipole to its antipode (*c.f.* the lines of longitude between the north and south poles of the earth): given the 3D line joining the two camera centres, take the circle of 3D half planes with this line as edge and project each half plane (on edge) into the two images making its two epipolar half-circles; the image of any 3D point on the half plane lies on both half-circles. In terms of a flat image, there is thus a full circle of distinct epipolar half-lines, and for any given point $\mathbf{x}'$, only the correct half of its epipolar line (to the left of the epipole or to the right of it) need be searched for its correspondent $\mathbf{x}$. Hence, this framework provides slightly sharper epipolar constraints whenever the epipole is in the image (notably for omnidirectional cameras).

Algebraically, with appropriate choices of signs, we can write this as $\mathbf{F}\mathbf{x}' \sim [e]_\times \mathbf{x}$ with equality up to positive rescalings. Using (3) and cancelling a common factor of $\mathbf{B}^\top \left(\begin{smallmatrix} 0 & 1 \\ -1 & 0 \end{smallmatrix}\right)$, we find that $\mathbf{B}'\mathbf{x}' \sim \mathbf{B}\mathbf{x}$, again with equality up to positive rescalings. So the epipolar pencil coordinates $(\alpha,\beta)^\top$ fully support the circular structure: the coordinates of corresponding points coincide up to *positive* rescalings and we can view them as angular coordinates $(\cos\theta,\sin\theta)^\top$ with $0 \le \theta < 2\pi$ and correspondence possible if and only if $\theta = \theta'$. The necessary signs for $\mathbf{B},\mathbf{B}'$ can not be recovered from $\mathbf{F}$ alone[10] so the easiest approach is to take a pair of corresponding spherical points, check that their pencil coordinates correspond, and if not flip the sign of $\mathbf{B}$. This will suffice for us here, although further refinements to ensure right-handedness, put $\mathbf{u},\mathbf{v}$ in canonical positions, *etc.*, are possible.

We can extend this to epipolar pencil conic matching. The basic formulae for conics and Kruppa constraints are intrinsically unsigned because converting $\left(\begin{smallmatrix} c \\ 1 \end{smallmatrix}\right)$ to its antipode $\left(\begin{smallmatrix} -c \\ -1 \end{smallmatrix}\right)$ leaves $\mathbf{q}$ unchanged in (4). This is reflected in the systematic appearance of $2\bar{\theta}$ in (10), *etc.* To correct for this we can algebraically 'unwrap' the first two rows of (11). Using multiple angle formulae and assuming normalization to $p^2+q^2=1$, we have

$$\begin{pmatrix} \cos\bar{\theta} \\ \sin\bar{\theta} \end{pmatrix} = \pm \begin{pmatrix} \sqrt{(1+p)/2} \\ q/\sqrt{2(1+p)} \end{pmatrix} = \pm \begin{pmatrix} q/\sqrt{2(1-p)} \\ \sqrt{(1-p)/2} \end{pmatrix} \tag{14}$$

[10] $\mathbf{F}$ itself can usually only be recovered up to sign, and even then the SVD of $\mathbf{F}$ is invariant under $(\mathbf{u},\mathbf{v}') \leftrightarrow (-\mathbf{u},-\mathbf{v}')$, $(\mathbf{v},\mathbf{u}') \leftrightarrow (-\mathbf{v},-\mathbf{u}')$, $\mathbf{e} \leftrightarrow -\mathbf{e}$, and $\mathbf{e}' \leftrightarrow -\mathbf{e}'$, some of which affect $\mathbf{B},\mathbf{B}'$.

where for numerical stability the first form is preferred if $p > 0$ and the second if $p < 0$. To choose the sign we take any signed point within (the desired branch of) the ellipse – *e.g.* its centre $\binom{c}{1}$ – find its pencil projection $\mathbf{Bx}$ or $\mathbf{B}'\mathbf{x}' \sim \binom{\cos\theta}{\sin\theta}$, choose the sign that aligns (14) the best with this direction, and replace the first two coordinates of (11) with the result (14) so that the feature vector encodes the desired half of the epipolar line. The right hand side of (12) is unchanged but it now represents only $\sin^2(\bar{\theta} - \bar{\theta}')$ so it should be scaled up by 4 to preserve the relative weightings of position and scale errors.

References

[1] P. Bendale, B. Triggs, and N. Kingsbury. Multiscale keypoint analysis based on complex wavelets. In *British Machine Vision Conference*, 2010.

[2] O. Chum, T. Werner, and T. Pajdla. Joint orientation of epipoles. In *British Machine Vision Conference*, 2003.

[3] O. Faugeras, Q.-T. Luong, and S.J. Maybank. Camera self calibration: Theory and experiments. In *European Conference Computer Vision*, 1992.

[4] P-E. Forssén and A. Moe. Blobs in epipolar geometry. In *SSBA Symposium on Image Analysis*, pages 82–Ű85, 2004.

[5] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003. ISBN 0521623049.

[6] F. Kahl and A. Heyden. Using conic correspondence in two images to estimate the epipolar geometry. In *IEEE International Conference on Computer Vision*, 1998.

[7] S. Laveau and O. Faugeras. Oriented projective geometry for computer vision. In *European Conference Computer Vision*, pages 147–156, 1996.

[8] D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal Computer Vision*, 60(2):91–110, 2004.

[9] K. Mikolajczyk and C. Schmid. Scale and affine invariant interest point detectors. *International Journal Computer Vision*, 60(1):63–86, 2004.

[10] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. J. Van Gool. A comparison of affine region detectors. *International Journal Computer Vision*, 65(1-2):43–72, 2005.

[11] B. Triggs. Autocalibration and the absolute quadric. In *IEEE Conference Computer Vision & Pattern Recognition*, 1997.

[12] B. Triggs. Detecting keypoints with stable position, orientation and scale under illumination changes. In *European Conference Computer Vision*, 2004.

[13] S. Winder, G. Hua, and M. Brown. Picking the best DAISY. In *IEEE Conference Computer Vision & Pattern Recognition*, pages 178–185, 2009.

Appendix C

Cambridge toy cars dataset

The images were captured using two Nikon D40 cameras with standard 18–55mm lenses and a pan-tilt unit (PTUD46 from Directed Perception) on a Unix system. The pan-tilt unit handles the rotation of the turn-table. Both the pan-tilt unit and the cameras are controlled remotely using a shell script or the command prompt. More information about related commands may be seen in gPhoto2 [Mueller et al., 2000] command reference and PTU D46 user manual at [Perception, 2006]. The shell script used to capture data (*i.e.* interact with PTUD46 and the cameras) is available at the web site for the dataset and in [Bendale et al., 2010a]. A few fairly basic example routines illustrating the use of the dataset for point transfer and the associated ground truth are available on the web site. Example images (cropped suitably to show the cars) are shown in Figure C.1.

C.1 Calibration

A two step calibration process is followed. In the first step, we use the CalDe/CalLab software [Strobl et al., 2005] to obtain the internal parameters of the camera as well as the rotation and translation between the two cameras. In the second step, we recover the rotation parameters for transferring points between two views of the same camera.

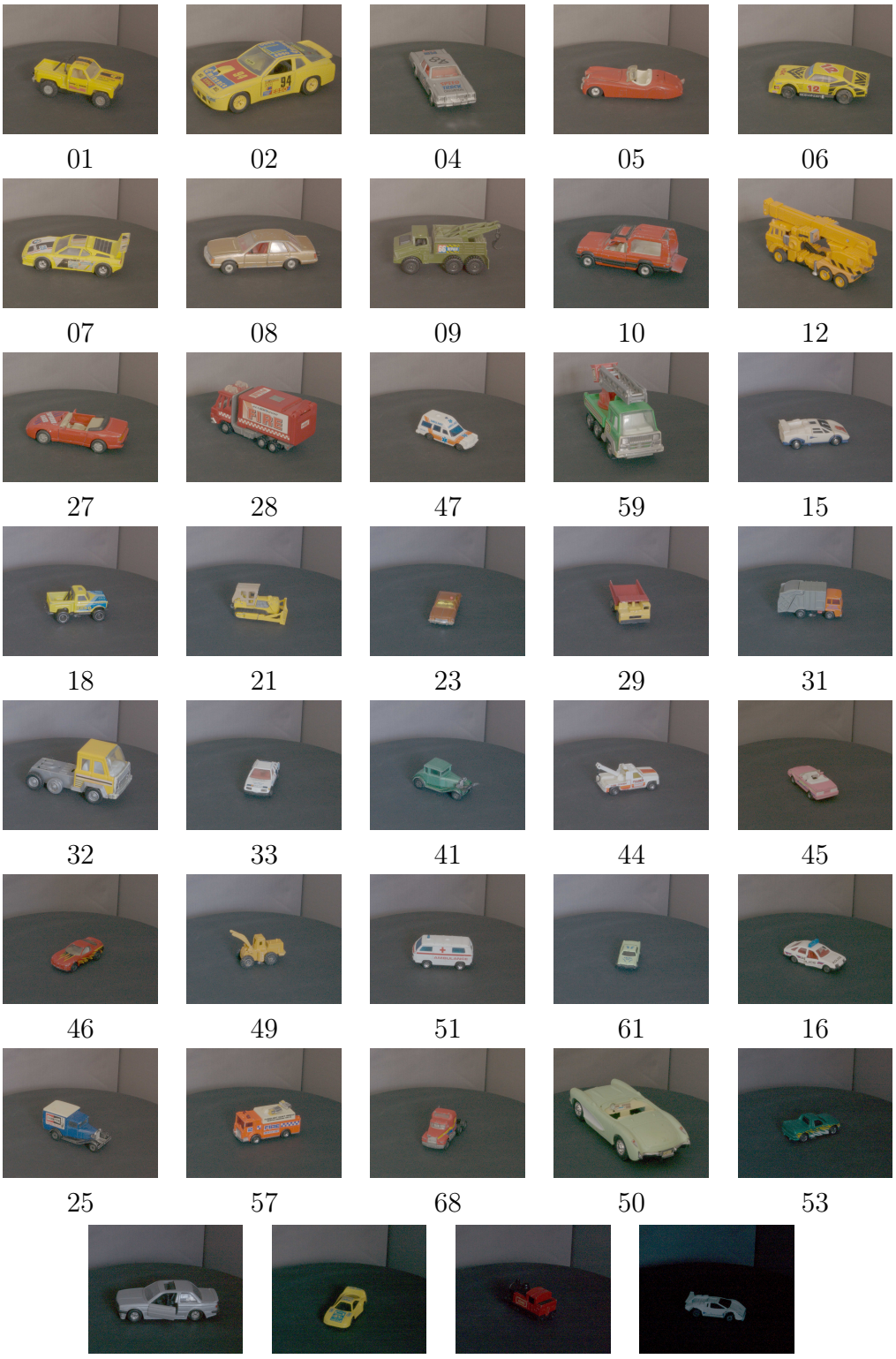


Figure C.1: Central 1024 tall ×1536 wide patch from each image in the dataset.

C.1.1 Inter-camera calibration

The radial distortion model used in DLR CalDe/CalLab toolbox [Strobl et al., 2005] (which has been used for the calibration of our dataset) can be expressed as

$$\begin{aligned} x' &= x + k_1 (x - x_0) \hat{r}^2 + k_2 (x - x_0) \hat{r}^4 + \ldots \\ y' &= y + k_1 (y - y_0) \hat{r}^2 + k_2 (y - y_0) \hat{r}^4 + \ldots \end{aligned} \tag{C.1}$$

where k_1 and k_2 are radial distortion coefficients, $\hat{r}$ is the observed radial distance of a point measured as

$$\hat{r}^2 = \left(\frac{(x - x_0) - s\,(y - y_0)/f_y}{f_x} \right)^2 + \left(\frac{y - y_0}{f_y} \right)^2 \tag{C.2}$$

and (x', y') are the predicted perspective projections of (x, y). The distortion is due to the camera lens so it happens after perspective projection but before digitizing.

While converting from actual image positions to ideal image positions, it is necessary to undo the effect of lens distortion (*i.e.* undistort the coordinates), whereas while converting from ideal image positions to actual image positions, it is necessary to apply appropriate lens distortion (*i.e.* distort the coordinates). Ideal image positions obey all rules of linear projection (*i.e.* epipolar lines are straight lines). Lens distortion has the effect of converting these into curves diverging away from the optical centre in the actual image positions. Lines in the real world do not get imaged as lines. The effect is not negligible if one wishes to establish sub-pixel correspondences. The extent of the lens distortion is shown in Figure C.2.

C.1.2 Rotational calibration

In the second step, we fit an ellipse to the world points obtained by triangulating the points on the edge of the turn-table in images from both the cameras. Points on the edge of the turn-table are selected in one image, an epipolar line is projected in the other image, then the correspondence (inter-

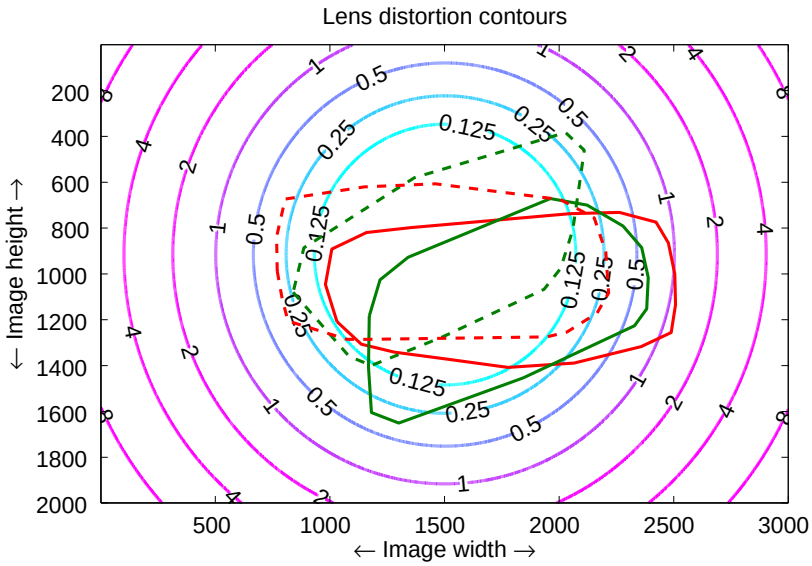


Figure C.2: Contours of lens distortion over actual image dimensions. Polygonal boundaries of two of the biggest cars in the dataset are shown in red (car 02) and green (car 12) lines. The dotted lines show the polygonal boundaries in the images from the upper camera (auxiliary images). The outer edges and far corners have significant lens distortion. All the cars occupy the central portion of the image with lens distortion ≤ 1 pixel.

section of the epipolar line and the edge of the turn-table) is marked manually on the epipolar line. These correspondences are then triangulated using the existing inter-camera calibration to obtain world point coordinates. This has to be done in only one view because the turn-table rotates around the central point, but the boundary of the turn-table is fixed in all views. Using the least squares approximation of the true boundary of the turn-table, we obtain the centre as well as the direction of the axis of rotation of the turn-table. The estimated location of the centre and the axis of rotation of the turn-table is shown in Figure C.3. These points are clicked manually. Therefore, it would be advisable to leave a fairly small (but obvious) marker on the turn-table to aid rotational calibration. Further details of the calibration and data capture process can be found in [Bendale et al., 2010a].

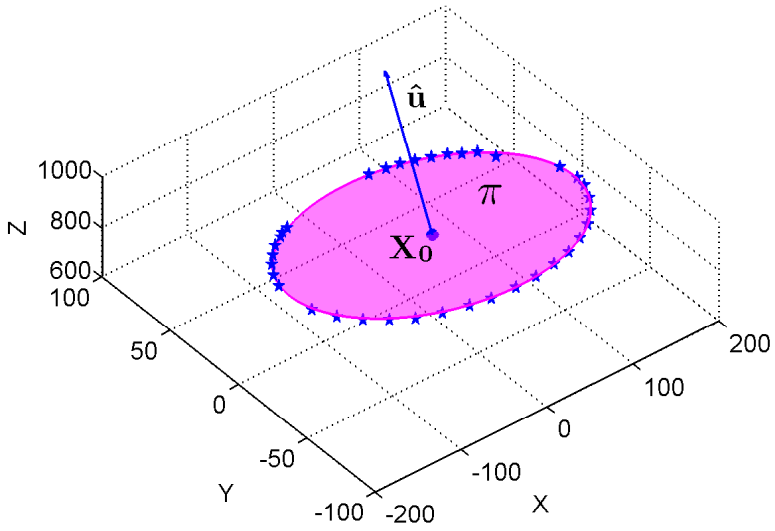


Figure C.3: The blue stars are the world points obtained by triangulating the points clicked on the edge of the turn-table in images from the lower camera and upper camera. The magenta line is the least squares approximation of the edge of the turn-table. The plane π is the plane of the top of the turn-table. The point $\mathbf{X_0}$ is the centre of the turn-table and the blue arrow indicates the direction of the axis of rotation. Unit vector in the direction of axis of rotation of the turn-table is $\hat{\mathbf{u}}$. Note that the turn-table is not parallel to the $X-Y$ plane because the cameras look down on the turn-table at an angle (lower camera: 30°, upper camera: 45°), and the centre of the lower camera is the origin of the coordinate system.

C.1.3 Extensions

- For this dataset, the camera batteries had to be removed for charging after capturing data for about 15 cars. A mains adapter would avoid this requirement.
- Proper diffuse photographic lighting would improve image quality and consistency.
- A textured cover for the turn-table top would be interesting. Our turn-table is supported on a pan-tilt unit, allowing controlled tilts of the test surface to be introduced. That would allow separation of 3D evaluation and 2D evaluation with the same setup [Fitzgibbon, 2010].

Appendix D

Cluster-Cluster matching

A manuscript[1] submitted to the IEEE Computer Society Conference on Computer Vision and Pattern Recognition [CVPR], 2008 under the title

> Pashmina Bendale, James D B Nelson, Nick Kingsbury, "Techniques for Establishing Keypoint Correspondences via Polar Matching with Complex Wavelets"

is included in the following eight pages and indicates status of the said work as of 10 December 2007.

[1]Pashmina Bendale wrote the software, planned and performed the experiments and wrote the text, Nick Kingsbury conceived the idea for section 7, provided software for the Dual-tree Complex wavelet transform and provided general supervision for rest of the work. James Nelson provided helpful comments.

Techniques for Establishing Keypoint Correspondences via Polar Matching with Complex Wavelets

Pashmina Bendale, James D. B. Nelson, Nick Kingsbury
Signal Processing Lab
Department of Engineering, Cambridge CB2 1PZ

Abstract

This paper illustrates a keypoint matching scheme for object recognition using the polar matching matrix descriptor. The polar matching matrix descriptor is a new rotation-invariant keypoint descriptor based on the Dual Tree Complex Wavelet Transform (DTCWT). The DTCWT basis functions allow us to locally determine scale, frequency, and orientation for a given image feature. Like SIFT, we detect keypoints spatially and in scale. However, instead of imposing a dominant orientation for every keypoint, we efficiently compute a confidence measure that a pair of keypoints match each other for the full range of orientations. We then use a clustering scheme that boosts weak matches that agree on pose and rejects false alarms. Furthermore, we demonstrate that our clustering scheme helps circumvent a winner-takes-all situation during the keypoint matching stage. Lastly, we show how this can be done in a fast and efficient way.

1. Introduction

A keypoint is essentially a two dimensional structure in an image that is likely to attract visual attention. Various recent approaches to object detection [2] and recognition [7, 20, 19] have concentrated on local-feature based methods because they are robust to occlusion and clutter, non-planar regions can be approximated as planar regions [23] and loose global geometric constraints can be incorporated at a local computational cost. This enables us to get global inference at local cost.

A common approach to interest point based object recognition is to first detect an interest point in location, scale and orientation space. A descriptor of the keypoint neighbourhood is then constructed and used to search for matching interest points in a new image.

2. Orientation–to keep or not to keep?

Some approaches to feature extraction, *e.g.* SIFT, build several features at the same location with multiple orientations [16], to achieve rotation-invariance, and thereby describe some parts of the image multiple times. When such keypoints are matched, the dominant orientation may govern the match probability. Therefore, there is a possibility that errors in dominant orientation estimation propagate to the matching stage. Since only one orientation can be matched reliably, the matching output may be unpredictable in the case of features that have self-symmetries.

Other rotation invariant approaches include spin images for object recognition [9] and texture classification [14] and Gabor-like filter banks for texture classification [27] and content based image retrieval [22]. Filter bank approaches to feature extraction as well as texture classification have concentrated largely on isotropic filters like Gaussian, Laplacian of Gaussian (LoG) or Difference of Gaussian (DoG) [25, 16, 27]. In cases where oriented filters are used, the maximum response across all directions is used at each scale [28, 15, 25] but this ignores all other angular statistics in order to achieve rotation-invariance.

We propose an alternative technique for rotation-invariant feature extraction which preserves all angular information. Specifically, we refrain from assigning a dominant orientation to a keypoint during the keypoint detection or description stage. Instead, we estimate the relative location and orientation of the keypoint in the search image relative to that in the reference image at the matching stage. This approach allows us to consider all possible symmetries or rotations of the keypoints, and thus achieves a softer matching process and avoids a winner-takes-all situation.

Groups of keypoints often occur together in objects across various viewpoints. A recently proposed approach [18] incorporates this global information by augmenting the SIFT descriptor with a Shape Context [4] (based on edges) and weighting these two descriptors to achieve a flexible local-global tradeoff. In contrast, rather than match each individual keypoint, we propose a way to match clusters

of keypoints and show that this can lead to improved performance. Cluster-cluster matching achieves a more robust measure of the similarity of regions as it is potentially sensitive to occlusion. Also, we do not assume a universal local-global tradeoff for entire image; the cluster-cluster matching constraints can be made as loose or as tight as possible depending on the application. For instance, a bag-of-features implementation can be achieved with a loose cluster-cluster matching constraint and a part based model can be realised by a tightly constrained cluster-cluster matching process.

3. The Dual Tree Complex Wavelet Transform

Two essential requirements of object recognition are invariance and discriminability. The Dual-Tree Complex Wavelet Transform (DTCWT) introduced by Kingsbury in [11], achieves invariance and discriminability for scale, location and orientation because the basis functions used in this transform are local in scale, space and orientation.

The DTCWT uses two trees of filter banks to obtain a real and an imaginary component of each of the wavelet coefficients. Since the filters in the two trees of the DTCWT are Hilbert Transforms of each other, we get an analytic and directionally selective output. Throughout the paper, the wavelet coefficients are denoted by $H\{k\}(d)$ and scaling coefficients by $L\{k\}$, where k is the DTCWT level and d is the subband direction. k takes values $(1, \ldots, N)$ for an N level DTCWT and d takes values $(1, \ldots, 6)$. The following properties of the DTCWT make it an attractive choice for the problem at hand:

1. **Approximate shift invariance:** DTCWT coefficients for any shift of an image can be approximately estimated by smooth interpolation of the DTCWT coefficients in each subband independent of all other subbands.
2. **Directional selectivity:** Orientations of image features can be accurately estimated from the DTCWT coefficients. This is because the scaling functions in the two trees of DTCWT form an approximate Hilbert pair making the transform analytic and hence directionally selective [24].
3. **Separability:** A very efficient separable filter bank implementation is available for the DTCWT.

The DTCWT filter responses are similar to those of a 6-directional Gabor transform with orientations of $\pm 15°; \pm 45°; \pm 75°; \pm 105°; \pm 135°; \pm 165°$, but the DTCWT is implemented using real filters. The price of this is limited redundancy of $(2^m{:}1)$ for m-dimensional signal (4:1 for images). A more detailed mathematical analysis of the Dual Tree Complex Wavelet Transform (DTCWT) is [12, 24].

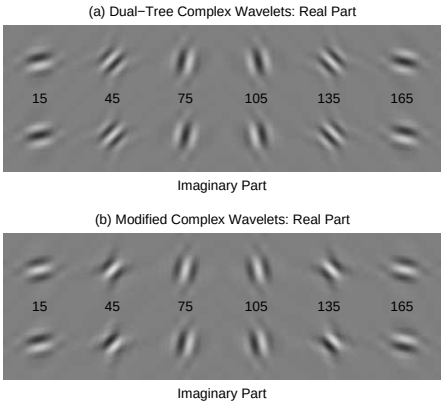


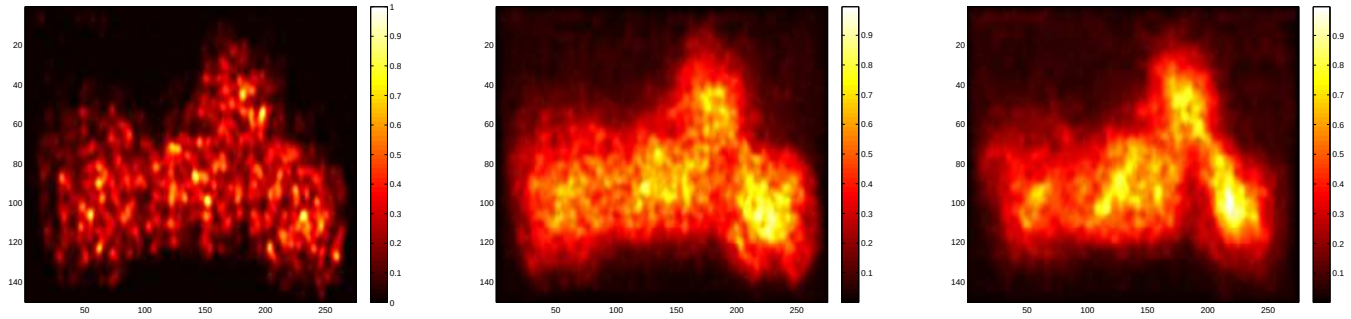
Figure 1. Impulse responses of DTCWT before and after addition of an extra bandpass filter in each dimension and phase correction to have zero phase at the mid-point of the responses. This results in a more rotationally symmetric DTCWT. Figures reproduced from [13].

3.1. DTCWT with improved rotational symmetry

Although the DTCWT as described in [12] has attractive perfect reconstruction properties, it is not rotationally symmetric. One alternative is to use the Steerable Pyramid [26] because it is rotationally symmetric but we concentrate on DTCWT here because of it's greater computational efficiency. The $45°$ and $135°$ subband centre frequencies of the DTCWT are further away from the origin than the other four subbands at a given scale in the frequency spectrum. This is because the centre of the 1D Hi filter is thrice as far than the 1D Lo filter (because they both span half the bandwidth of the input signal), so a 2D Lo-Hi filter formed from a combination of 1D Lo and 1D Hi is closer to the origin than a 2D Hi-Hi filter by a factor of $\sqrt{3^2+3^2}/\sqrt{3^2+1^2} = \sqrt{1.8}$.

For feature description, the perfect reconstruction constraint can be relaxed to create a more rotationally symmetric version of the DTCWT. Kingsbury [13] suggested that an additional bandpass filter may be added in each dimension to pull the $45°$ and $135°$ subband centre frequencies closer to the origin by $\sqrt{1.8}$.

Another feature of the standard DTCWT is that all six subbands may not have zero phase at the mid-point of their responses. As in [13], a phase correction of $\{j, -j, j, -1, 1, -1\}$ has been applied in the new DTCWT version to make all real parts of the six subband responses even symmetric and imaginary parts of all the six subband responses odd-symmetric. This property allows one to calculate responses in the opposing directions $(30d - 15 + 180)°$ by conjugating the responses of the original six subbands, $(30d - 15)°$. The orientation of zero crossing changes in a cyclic manner across the six subbands. The phase-corrected impulse responses are compared in Figure 1.



(a): DTCWT detector [6] (b): Enhanced DTCWT (EnDTCWT) detector (c): DoG detector

Figure 2. Keypoint repeatability test as illustrated by Kadir *et al*. [10]. Smoothed maps of the three keypoint detectors over 200 images accumulated in one image. The colour indicates the normalised number of detections in a given area (white is highest). Enhanced DTCWT keypoint detector detects repeatable keypoints over a larger proportion of the bike than the DTCWT keypoint detector (a) and Difference of Gaussian (DoG) (c). Also, DTCWT keypoints are seen to be fairly insensitive to background clutter. 84 images with a uniform background and 116 images with varying degrees of background clutter are used from the CALTECH motorbikes (side) dataset as in [10].

3.2. DTCWT keypoint detector

Our approach to keypoint matching is based on the keypoint detector recently introduced by Fauqueur *et al*. [6]. The DTCWT keypoint detector detects local interest features at locations characterised by large magnitudes of wavelet response in all or most of the six subband directions. Products of the magnitudes of wavelet coefficients in six directions are considered at each location in the image. This product is large for corners (most of the six values are large as there is considerable leakage between the subbands) and for blobs (all six values equal and large). Edges have low response in a few subbands and hence points on edges are not picked as keypoints.

To make the approach multi-scale, products are calculated at every level and the product maps are combined using an amplitude accumulation process [6]. Accumulation boosts features that show consistency over a range of scales. Noise is characterised by random statistics and hence does not show any strong structure across scales. The product map $M_\Pi\{k\}(x,y)$ at scale k and location (x,y) is calculated from wavelet coefficients of the luminance component using,

$$M_\Pi\{k\}(x,y) = \left(\prod_{d=1}^{6} |H\{k\}(x,y,d)|\right)^{1/4} \quad (1)$$

Keypoints are marked at local maxima in this accumulated product map. DTCWT subband responses can be smoothly interpolated at intermediate directions because there is leakage across adjacent subbands. The scale of a keypoint is detected at the distance where the radial gradient of the accumulated map in eight equi-spaced directions has a strong minimum. This distance is found by projecting rays outward from the keypoint location in eight directions (multiples of $45°$) and calculating gradients along these rays using a forward difference. The distance at which the sum of negative gradient over all directions reaches a maximum is marked as the scale of the keypoint. For robustness to noise in the accumulated map, the minimum is detected using an area accumulation technique.

4. Enhanced DTCWT keypoint detector

Our keypoint detector is based on [6], but we have enhanced it as follows. We use the root mean square(RMS) wavelet amplitude across RGB components instead of using luminance. We contrast-equalise the wavelet coefficients before keypoint detection and also locate the keypoints with subpixel accuracy. Further, we use a slightly different formula to compute the product map. (Equation 2) We use a value of 1/6 for exponentiation, instead of 1/4 in [6], because it preserves dimensionality and direct proportionality through the product operator. Thus, if all values are scaled by a factor α, the product map also gets scaled by a factor α. This operation then becomes a geometric mean of the six subband responses and has maximum value when all values are equal.

$$M_\Pi\{k\}(x,y) = \left(\prod_{d=1}^{6} E_{rms}\{k\}(x,y,d)\right)^{1/6} \quad (2)$$

$$E_{rms}\{k\}(x,y,d) = \sqrt{\sum_{c=RGB} |H_c\{k\}(x,y,d)|^2} \quad (3)$$

We find that keypoints detected using this keypoint detector are more stable and more characteristic of the image content compared with the detector in Equation 1. Furthermore, it is capable of picking low contrast locations with sufficient colour activity as keypoints. Computational complexity is similar to [6], since we are still using only one channel, i.e the RMS amplitude but there is a useful improvement in the performance for colour images with large contrast changes within the image. Figure 2 shows a comparison of the two detectors.

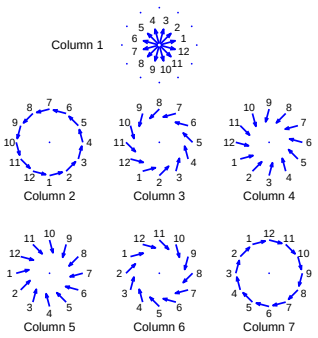


Figure 3. Polar matching matrix formation: Each arrangement describes the formation of one column of the Polar matching matrix. The numbers associated with the arrows in each column arrangement give the row index of the subband in the corresponding column of the Polar matching matrix. The direction of the arrows decides the subband used and the location of the base of the arrow in each arrangement decides which point of the 13 points is used to create the Polar matching matrix. Figures reproduced from [13].

An accumulated map is derived from the $M_{\Pi}\{k\}(x, y)$ values at each scale k as in [6]. We improve the keypoint localisation by implementing a Hessian based interpolation in a 3×3 neighbourhood of the maxima to determine subpixel locations of the keypoints. The curvature of quadratic fit depends on the ratio of the largest to smallest eigenvalues, $r = \lambda_{max}/\lambda_{min}$ of the Hessian $\mathbf{H}$ evaluated at the pixel closest to the maxima. In the vicinity of a maximum, $\mathbf{H}$ is symmetric and positive definite, hence the eigenvalues satisfy $0 \leq \lambda_{min} \leq \lambda_{max}$. The most dominant keypoints are detected when $r \simeq 1$. We find that an acceptance threshold value requiring $r \geq 0.1$ works well.

Prior to keypoint detection, we propose a contrast equalisation scheme, based on the wavelet coefficient energies at various levels, to enhance the keypoint detection ability. At a given location, the total wavelet energy content of the signal is a measure of the contrast. To measure contrast variations that persist across scales, the total activity energy at a point at each level is calculated using the wavelet energy at the point, its four children from one level below, and the four children of each of these from two levels below the parent level. The contrast correction factor is then calculated on basis of this activity energy, such that within limits the resulting image has roughly equal contrast throughout the image. Since both the real and imaginary parts of all three wavelet coefficients H_R, H_G, H_B are scaled by the same factor, the phase and color balance of wavelet coefficients is preserved.

5. Polar matching matrix

We use a new rotation invariant keypoint descriptor introduced by Kingsbury [13]. This descriptor uses the modified DTCWT version, explained in Section 3.1. The

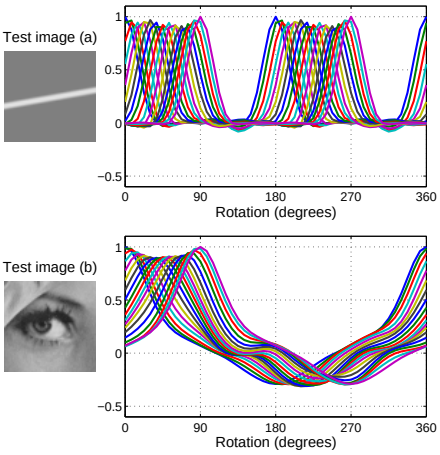


Figure 4. Polar matching matrix correlation scores vary smoothly as an image is rotated. In case of self-symmetric features, (like the bar edge) the correlation curve shows exactly as many peaks as the degree of self-symmetry of the feature. In case of asymmetric features like *Lena's eye* the correlation curve has only one well defined peak and correlation scores are strongly negative at some angles indicating a non-match. It is worth noting that the dip in the peak correlation value is very small for different orientations of the object. Figures reproduced from [13].

DTCWT coefficients within any given subband are sufficiently band-limited to allow us to interpolate between them smoothly at any desired sampling locations. Descriptors of arbitrary 'richness' can thus be built using this property. We describe the 2-scale single ring version here.

DTCWT produces responses in six directions $(30d - 15)^\circ$ for $d = (1, 2, \ldots 6)$, spanning the first two quadrants of the frequency plane. The third and fourth quadrant are mirror images of the first and second quadrant and hence the responses in the six opposing subbands at $(30d-15+180)^\circ$ for $d = (1, 2, \ldots 6)$ can be obtained by conjugating the responses of the six actual subbands. Thus there is information in 12 directions, as shown in Figure 3. The wavelet responses in 12 directions at 12 points on a circle centred at the keypoint and those at the centre are used to form the descriptor matrix, called the Polar matching matrix, $\mathbf{P}$ ($\mathbf{P}$-matrix). The point at the centre is marked M, that at the 9-o-clock position is marked A, continuing in a clockwise order until L is at the 8-o-clock position. Further details of $\mathbf{P}$-matrix construction are in [13, 3]. The Polar matching matrix $\mathbf{P}$, has the following attractive properties from the point of view of object recognition:

1. **Rotation invariance, Rotation estimation:** Rotation of the object around the keypoint produces cyclic shifts in the columns of the $\mathbf{P}$-matrix.
2. **Illumination and contrast invariance:** The $\mathbf{P}$-matrix uses band-pass wavelet coefficients and its energy is normalised before being used for matching. Hence it is unaffected by pixel intensity offsets and scaling.

The $\mathbf{P}$-matrix can be extended to multiple scales by adding more columns to the right of the seventh column. It is also possible to have more than one ring of sampling points. The key is to use a radius and therefore a region just big enough to include enough information about the keypoint neighbourhood to distinguish it from other keypoints and yet not take much background information.

In tandem with our colour keypoint detection method, we have extended this work to colour images. Although the colour values of a keypoint location may vary significantly due to lighting variations, the colour ratios should remain fairly constant. If three separate matrices, $\mathbf{P}_R, \mathbf{P}_G, \mathbf{P}_B$ are built for each colour channel and combined into a single matrix, normalised by the total energy, then colour information can be used for matching in a robust way. Note that other colour spaces (*e.g.* Lab) could be used here, but we find that RGB works well enough. An alternative is to use the colour information only for the midpoint and luminance for the sampling circle.

The $\mathbf{P}$-matrix correlation scores are highly invariant to lighting changes but they are sensitive to small errors (a few pixels) in keypoint location errors. To correct for small errors, a shift correction [3] for the keypoint location is applied by moving the keypoint location in a direction which maximises the correlation score. This direction is estimated by measuring the derivatives of the Polar matching matrix with respect to small shifts of the sampling circle in x and y direction. The correlation scores for each location at 48 rotations are then estimated by using a least mean squares solution. This results in a smoothly varying estimate of correlation scores and hence greater tolerance to shifts in keypoint locations. This is described in further detail, in a companion submission [3].

6. FFT based Polar matching matrix correlation

An FFT based matching scheme for Polar matching matrix descriptors was proposed in [13] that looks for cyclic shifts in the columns of the $\mathbf{P}$-matrix. $\mathbf{P}_{s,j}$ denotes the $\mathbf{P}$-matrix for the j^{th} keypoint in the search image and $\mathbf{P}_{r,i}$ denotes the $\mathbf{P}$-matrix for the i^{th} keypoint in the reference image.

The goal is to have a correlation result that tells us the strength of correlation for all possible rotations of the object. The location of the peak of the correlation curve thus formed can be used to estimate the angle of rotation. DTCWT subband directions have a spacing of $30°$ between them. Therefore 12 points were chosen on the sampling circle for $\mathbf{P}$-matrix. Fourier interpolation then creates a 48-point correlation curve at shifts of $7.5°$. The column-wise matching process proceeds as follows:

1. Compute FFT of $\mathbf{P}$-matrix for all keypoints in reference image: $\overline{\mathbf{P}}_{sj} = \mathrm{FFT}\{\mathbf{P}_{sj}\}$

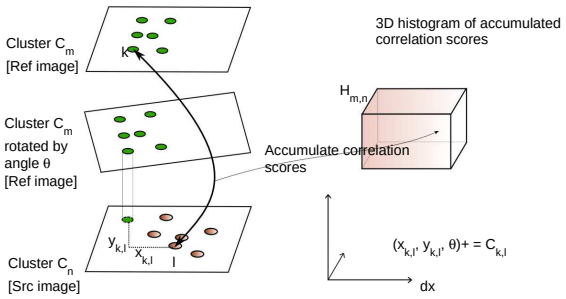


Figure 5. Matching clusters of keypoints between two images. Polar matching matrix correlation scores are accumulated at relative cluster displacements and orientations to determine correspondences.

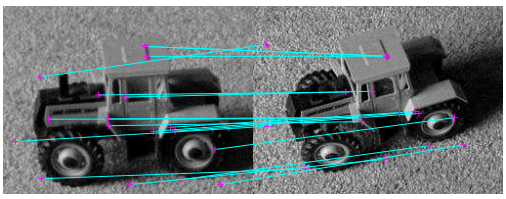
2. Compute FFT of $\mathbf{P}$-matrix for all keypoints in search image: $\overline{\mathbf{P}}_{ri} = \mathrm{FFT}\{\mathbf{P}_{ri}\}$
3. Compute pairwise products with corresponding conjugates $\overline{S}_{i,j} = \overline{\mathbf{P}}_{ri} \cdot \{\overline{\mathbf{P}}^*_{sj}\}$ using element wise multiplication, where $\overline{\mathbf{P}}^*$ denotes conjugation of matrix $\overline{\mathbf{P}}$.
4. Zero-pad the low-energy parts of the pairwise products, $\overline{S}_{i,j}$, to get $\overline{s}_{i,j}$. This avoids aliasing and leads to good interpolability as pointed out in [13].
5. The search and reference images used are both real, therefore we need to use only the real part of the Inverse Fourier Transform (IFFT) of $\overline{s}_{i,j}$ signal, to get the 48 point correlation result, $s_{i,j}$.

The peak in the correlation curve $s_{i,j}$ gives the orientation and score of the best match. Steps 1 and 2 are done only once per keypoint in each image. Steps 3-6 are done for each combination of pairs of keypoints between the two images.

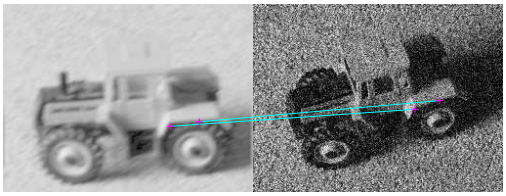
7. Cluster-Cluster correspondences

Based on our robust keypoint detector and the Polar matching matrix descriptor, we propose a new cluster-wise keypoint matching scheme. We briefly mention our cluster matching scheme here (Figure 5) and show preliminary results (Figure 8). Groups of keypoints often occur together over a range of viewpoints. We exploit this fact to aid pairwise keypoint matching to get the best possible keypoint correspondences, given a set of pairwise matching scores.

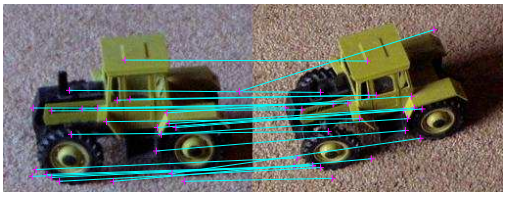
Small clusters of keypoints are formed across the entire image. Clusters are allowed to overlap each other, so every keypoint contributes to several nearby clusters. A cluster is picked in the reference image and all keypoints in the cluster are rotated about the centroid by the same $7.5°$ angle shifts used in the pairwise correlation calculation. For every rotation, θ between the reference image cluster and search image cluster, x and y location discrepancies $(\mathrm{d}x, \mathrm{d}y)$ are



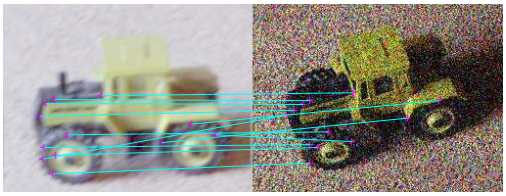
(a): SIFT 18 matches



(b): SIFT 2 matches



(c): EnDTCWT-SIFT 25 matches



(d): EnDTCWT-SIFT 10 matches

Figure 6. Comparison of the SIFT DoG detector and the Enhanced DTCWT keypoints described by SIFT descriptors. All figures in first row have non-degraded images. In each figure in second row, the left image is first motion blurred (size = 7) and then gamma distorted (γ = 0.3), right image is corrupted by Gaussian noise (σ^2 = 0.05). (a),(b): Images matched by SIFT system [D. Lowe's code from [16]]. (2 of 18 matches survive the degradation) (c),(d): Images matched with SIFT descriptors when keypoint locations are picked by our Enhanced DTCWT(EnDTCWT) keypoints. (10 of 25 matches survive the degradation). This shows that our enhanced DTCWT keypoint detector picks stronger keypoints which can be useful in object recognition from low-quality images.

computed for all pairs of keypoints within the search and reference image cluster. The pairwise correlation score between the keypoint pairs is then accumulated in a 3D histogram at a location $(\mathrm{d}x, \mathrm{d}y, \theta)$.

When a significant number of keypoints are true correspondences, they agree on the pose $(\mathrm{d}x, \mathrm{d}y, \theta)$, and contribute at the same locations in the 3D histogram producing a maximum at that pose. A peak in the 3D histogram is thus indicative of a cluster match. We use trilinear interpolation to bin the correlation scores into the histogram bins. A third degree surface fitting procedure is used to find the sub-pixel location of the true maximum. The location of the

Figure 7. Two stage keypoint matching results for two views of the *Tractor*. Left: Individual keypoint matches: Initial keypoint matching ignores all angular constraints and selects the best match as the correspondence. Right: Keypoint matches after cluster-wise matching. Some new matches are generated due to cluster matching (front of the tractor) and some false alarms are rejected (carpet and similar looking wheel).

maximum in the 3D histogram is the estimated pose of the search cluster relative to the reference image cluster. The value at the maximum is the matching score for the search and reference image cluster pair.

Keypoint correlation scores are rearranged with respect to the cluster matches and keypoint matches are found by picking the best keypoint match among all the keypoints of the matched pair of clusters. Once cluster-cluster matches have been established, the cluster matching is used to constrain the keypoint matching to only include points within the matched cluster pairs. During cluster matching, the angular information is used and hence weak matches of the first stage win in the second stage by virtue of being in the correct position and orientation though they might have produced a lower correlation score initially owing to viewpoint change. This results in fewer false alarms and more reliable matching results.

Complete orientation, location and scale information is preserved in our approach at every stage. Hence, the process of inferring cluster-cluster matching scores from keypoint-keypoint matching scores can be repeated at another stage to form clusters of clusters. Such a system will be characterised by some scale and shift invariance within each stage and greater invariance between consecutive stages making it suitable for a hierarchical object recognition system. There is evidence [17, 21] that the human brain performs hierarchical processing for object recognition. Hierarchical approaches to object localisation [1], object recognition [8, 25, 19] and object categorization [5] have consistently been shown to perform better than single stage approaches.

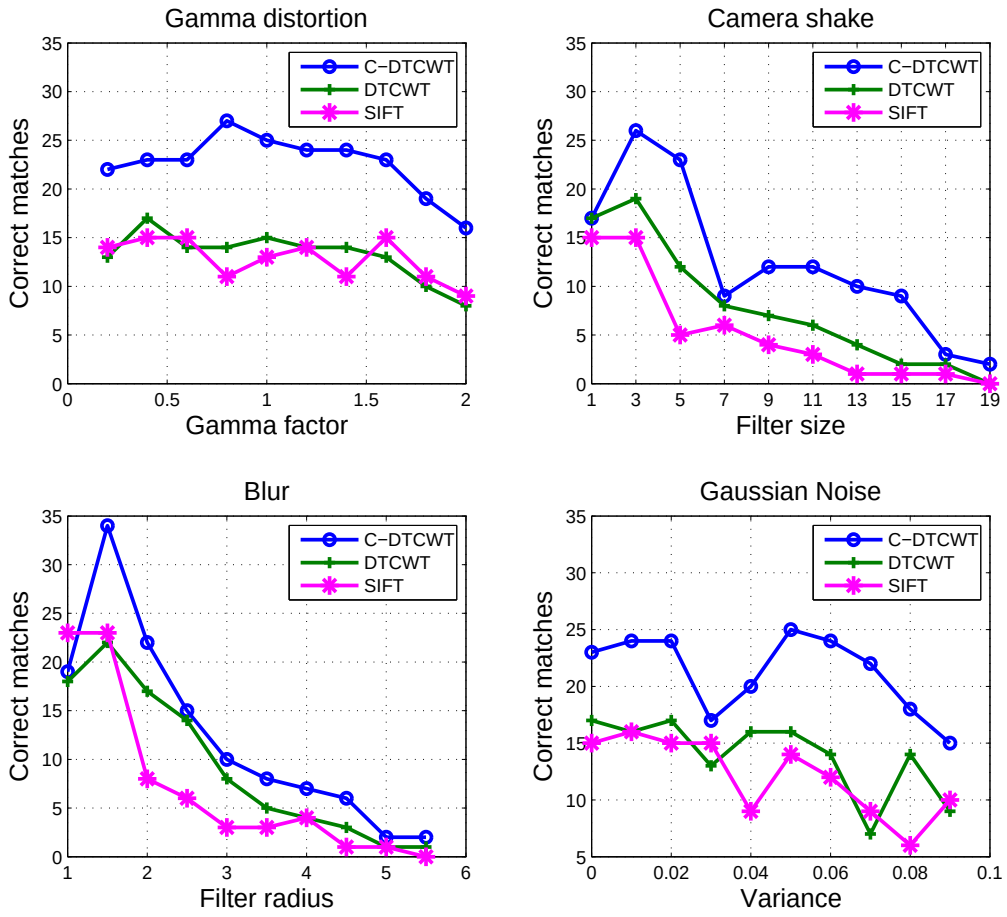


Figure 8. Quantitative evaluation of SIFT matching scheme, DTCWT individual keypoint matching and our cluster-wise DTCWT keypoint matching (C-DTCWT) for low quality images. Images used were the same as in Figure 7 and a single type of degradation is used for each test. The cluster-wise keypoint matching scheme shows improved performance over individual keypoint matching. In all tests, an image is compared with another image containing the same object from a different viewpoint and one of the images is degraded by the specified degradation.

8. Object recognition from low quality images

A quantitative evaluation of the SIFT matching scheme, DTCWT individual keypoint matching scheme and our cluster-wise DTCWT keypoint matching (C-DTCWT) scheme for low quality images is shown in Figure 8. We evaluate the three systems in presence of various image degradations like Camera shake, Gamma distortion, Blur and Gaussian noise that can typically be present in images. In the real world, most images have a combination of degradations, so we evaluate the three systems in presence of a composite degradation. The results can be seen in Figure 9. We see that our cluster matching gives more reliable keypoint matches than individual keypoint matching and can be useful in object recognition from low quality images as it uses weak spatial constraints along with orientation constraints.

9. Conclusion

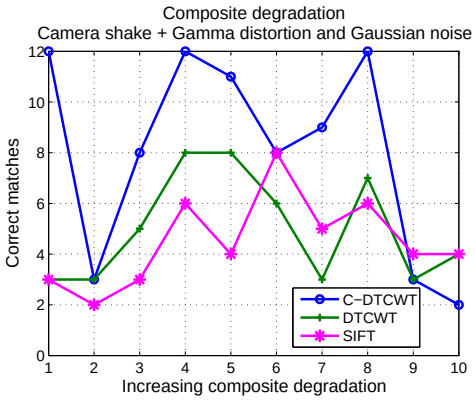
We have enhanced the keypoint detector and used it with a rotation invariant feature descriptor to obtain an orientation invariant confidence measure for keypoint correspondences. We have introduced a new cluster-wise matching scheme for keypoint matching. We show that this scheme reduces false alarms and encourages correct matches by enforcing weak spatial constraints.

References

[1] A. Agarwal and B. Triggs. Hyperfeatures - multilevel local coding for visual recognition. In *ECCV*, 2006. 6

[2] S. Agarwal and D. Roth. Learning a sparse representation for object detection. In *ECCV*, 2002. 1

[3] Authors. Displacement correction of the polar matching method for object detection, 2007. CVPR 2008 Submission



(a) Composite degradation

(b) Test images

Figure 9. (a): Performance evaluation in presence of a composite degradation. Increasing Camera shake and gamma distortion are applied to first image and increasing noise is added to second image. Camera shake filter size varies from 0.5 to 9.5 in 10 steps. Gamma factor varies from 0.2 to 2 in 10 steps for the Image A and noise variance varies from 0 to 0.1 in 10 steps. (b): Extreme cases of the test images used in the quantitative evaluation. Top-Left: Gamma distortion Top-Right: Gaussian noise Bottom-Left: Camera shake Bottom-Right: Uniform blur

ID 1204. 4, 5

[4] S. Belongie, J. Malik, and J. Puzicha. Shape context: A new descriptor for shape matching and object recognition. 2000. 1

[5] G. Bouchard and B. Triggs. Hierarchical part-based visual object categorization. In *CVPR*, 2005. 6

[6] J. Fauqueur, N. Kingsbury, and R. Anderson. Multiscale keypoint detection using the dual-tree complex wavelet transform. In *ICIP*, 2006. 3, 4

[7] I. Gordon and D. G. Lowe. What and where: 3d object recognition with accurate pose. In J. Ponce, M. Hebert, C. Schmid, and A. Zisserman, editors, *Toward Category-Level Object Recognition*, pages 67–82. Springer-Verlag, 2006. 1

[8] B. Heisele, T. Serre, M. Pontil, T. Vetter, and T. Poggio. Categorization by learning and combining object parts. In *NIPS*, 2001. 6

[9] A. E. Johnson and M. Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 21(5):433–449, 1999. 1

[10] T. Kadir, A. Zisserman, and M. Brady. An affine invariant salient region detector. In *ECCV*, 2004. 3

[11] N. G. Kingsbury. The dual-tree complex wavelet transform: a new technique for shift invariance and directional filters. In *IEEE DSP Workshop*, August 1998. 2

[12] N. G. Kingsbury. Complex wavelets for shift invariant analysis and filtering of signals. *Journal of Applied and Computational Harmonic Analysis*, 10(3):234–253, 2001. 2

[13] N. G. Kingsbury. Rotation-invariant local feature matching with complex wavelets. In *EUSIPCO*, 2006. 2, 4, 5

[14] S. Lazebnik, C. Schmid, and J. Ponce. A sparse texture representation using local affine regions. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(8):1265–1278, 2005. 1

[15] T. Leung and J. Malik. Representing and recognizing the visual appearance of materials using three-dimensional textons. *Int. J. Comput. Vision*, 43(1):29–44, 2001. 1

[16] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, pages 91–110, 2004. 1, 6

[17] D. G. Lowe and T. O. Binford. Perceptual organization as a basis for visual recognition. *AAAI-83*, pages 255–260, 1983. 6

[18] E. Mortensen, H. Deng, and L. Shapiro. A SIFT descriptor with global context. In *CVPR*, 2005. 1

[19] J. Mutch and D. G. Lowe. Multiclass object recognition with sparse localized features. In *CVPR*, pages 11–18, 2006. 1, 6

[20] S. Obdrzalek and J. Matas. Object recognition using local affine frames on distinguished regions. In *BMVC*, 2002. 1

[21] M. Riesenhuber and T. Poggio. Hierarchical models of object recognition in cortex. *Nature Neuroscience*, 2(11):1019–1025, 1999. 6

[22] C. Schmid. Constructing models for content-based image retrieval. In *CVPR*, 2001. 1

[23] C. Schmid and R. Mohr. Local grayvalue invariants for image retrieval. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(5):530–535, 1997. 1

[24] I. W. Selesnick, R. G. Baraniuk, and N. G. Kingsbury. The dual-tree complex wavelet transform. *IEEE Signal Processing Magazine*, 22(6):123–151, 2005. 2

[25] T. Serre, L. Wolf, and T. Poggio. Object recognition with features inspired by visual cortex. In *CVPR*, 2005. 1, 6

[26] E. P. Simoncelli and W. T. Freeman. The steerable pyramid: a flexible architecture for multi-scale derivative computation. In *ICIP*, 1995. 2

[27] M. Varma and A. Zisserman. Classifying images of materials: Achieving viewpoint and illumination independence. In *ECCV*, 2002. 1

[28] M. Varma and A. Zisserman. Texture classification: Are filter banks necessary? In *CVPR*, 2003. 1

Appendix E

Alternative maximum interpolation methods

E.1 Mean-shift scale estimation

A variable-bandwidth mean-shift process [Comaniciu et al., 2001, Comaniciu, 2003] is used to find maxima in scale-space. An application of the variable bandwidth mean-shift process to finding peaks in scale-space was described in [Dalal, 2006, Section 5.2]. We use a Gaussian kernel in both space and scale and the grid points are weighted by the keypoint responses obtained at these grid points. The 4S-DTCWT scale-space is of 'pyramidal' nature, *i.e.* has unequal sampling intervals (*c.f.* Figure. 3.1), the kernel width has to be scaled for each level according to the sampling interval for that level. The sampling interval of the level at which the candidate detection is localised (*i.e.* the scale value associated with the level) is used as the initial scale for each grid point as well as candidate detections. When the location of the candidate detection is updated, the kernel weights are recomputed and a new set of grid points is determined to be used in the mean-shift process. This process continues until convergence.

Assume that every grid point is represented by three coordinates (x, y, s) in the scale-space. All points in any given level k have the same scale s. Every grid point has a keypoint response w associated with it and the i^{th} detection is denoted as $\mathbf{x}_i = [x_i\ y_i\ s_i]^\top$. Here, s_i is the scale of i^{th} point on

$\log_2$ scale and s'_i is the scale of i^{th} point in pixels. The two are related by $s_i = \log_2(s'_i)$. The non-negative keypoint response at (x_i, y_i, s_i) is denoted by w_i. Assuming standard deviations σ_x, σ_y and σ_s in x, y and s, let

$$K(\mathbf{x}, \mathbf{x}_i, \mathbf{H}_i) \quad = \quad \exp\left[-\frac{1}{2}(\mathbf{x}-\mathbf{x}_i)^\top \mathbf{H}_i^{-1}(\mathbf{x}-\mathbf{x}_i)\right] \tag{E.1}$$

be the Gaussian kernel over all detections. The matrix $\mathbf{H}_i$ is the diagonal covariance matrix such that $\text{diag}(\mathbf{H}_i) = [(s'_i\sigma_x)^2\ (s'_i\sigma_y)^2\ (\sigma_s)^2]$.

We define a density function to describe the keypoint response distribution centred at the detection $\mathbf{x} = [x\ y\ s]^\top$ to be of the form,

$$\hat{f}(\mathbf{x}) \quad = \quad \sum_{i=1}^{n} w_i\, K(\mathbf{x}, \mathbf{x}_i, \mathbf{H}_i) \tag{E.2}$$

The gradient of this function is

$$\nabla\hat{f}(\mathbf{x}) \quad = \quad \sum_{i=1}^{n} \mathbf{H}_i^{-1}(\mathbf{x}-\mathbf{x}_i)\, w_i\, K(\mathbf{x}, \mathbf{x}_i, \mathbf{H}_i) \tag{E.3}$$

A mode being the local maximum, will be characterised by a zero gradient. At the modes, $\mathbf{x} = \overline{\mathbf{x}}$ and $\nabla\hat{f}(\mathbf{x}) = \mathbf{0}$, such that

$$\sum_{i=1}^{n} \mathbf{H}_i^{-1}(\overline{\mathbf{x}}-\mathbf{x}_i)\, w_i\, K(\overline{\mathbf{x}}, \mathbf{x}_i, \mathbf{H}_i) \quad = \quad \mathbf{0} \quad , \tag{E.4}$$

and the shift due to keypoint response in iteration $\tau+1$ is given by the mean-shift vector $\overline{\mathbf{x}}(\tau+1)$ as

$$\overline{\mathbf{x}}(\tau+1) \quad = \quad \frac{\sum_{i=1}^{n} \mathbf{H}_i^{-1}\mathbf{x}_i\, w_i\, K(\overline{\mathbf{x}}(\tau), \mathbf{x}_i, \mathbf{H}_i)}{\sum_{i=1}^{n} \mathbf{H}_i^{-1}\, w_i\, K(\overline{\mathbf{x}}(\tau), \mathbf{x}_i, \mathbf{H}_i)} \tag{E.5}$$

The mean-shift process takes as input a candidate detection and the keypoint response at various levels. A stable maximum in scale and space is found using the mean-shift process. The location of the mode can be affected by the irregular distribution of grid points and levels around the candidate detection and by the variation in density of grid points across levels (*i.e.* there

are more grid points at finer levels than there are at coarser levels). The bias introduced by irregular distribution of grid points and levels around the candidate position is equivalent to the shift the detection would undergo in a constant weight scenario (*i.e.* $\forall i, w_i = 1$). The bias, $\mathbf{b}(\tau+1) = \overline{\mathbf{x}}(\tau+1)|_{w_i=1}$

$$\mathbf{b}(\tau+1) \;=\; \frac{\sum_{i=1}^{n} \mathbf{H}_i^{-1}\,\mathbf{x}_i\,K(\overline{\mathbf{x}}(\tau),\mathbf{x}_i,\mathbf{H}_i)}{\sum_{i=1}^{n} \mathbf{H}_i^{-1}\,K(\overline{\mathbf{x}}(\tau),\mathbf{x}_i,\mathbf{H}_i)} \quad , \tag{E.6}$$

is subtracted from $\overline{\mathbf{x}}(\tau+1)$, to get the corrected estimate of the mean-shift vector,

$$\overline{\mathbf{x}}_c(\tau+1) \;=\; \overline{\mathbf{x}}(\tau+1) - \mathbf{b}(\tau+1) \tag{E.7}$$

We found that this method required samples from too many levels and the peak locations drifted in scale and space more than expected.

E.2 Adaptive Maximum Interpolation

This section describes joint work[1] with Nick Kingsbury and Hong Tao, Signal Processing Laboratory, Department of Engineering, Cambridge University.

A second order Taylor series approximation on a 3×3 neighbourhood is commonly used for estimating the value and location of the maximum of a function in image processing. Given a function $F(\mathbf{x})$ known at pixel locations, our aim is to find the location of the maximum, $\widehat{\mathbf{x}}$ and the function value $F(\widehat{\mathbf{x}})$. A differentiation of the second order Taylor series expansion of $F(\mathbf{x})$,

$$F(\mathbf{x}) \;=\; \mathbf{F} + \mathbf{g}^{\top}\mathbf{x} + \frac{1}{2}\mathbf{x}^{\top}\mathbf{H}\mathbf{x} \quad , \tag{E.8}$$

[1]Pashmina Bendale pointed out the problem, Nick Kingsbury conceived the solution, Pashmina Bendale formalised it, Hong Tao wrote the code for 4×4 maximum interpolation based on Nick Kingsbury's 3×3 maximum finder routine, Pashmina Bendale provided framework to test the 4×4 maximum interpolation, Hong Tao did the experiments to produce results in Figure E.2.

yields the expressions

$$\widehat{\mathbf{x}} = -\mathbf{H}^{-1}\mathbf{g} \quad \text{and} \tag{E.9}$$

$$F(\widehat{\mathbf{x}}) = \mathbf{F} + \frac{1}{2}\mathbf{g}^{\top}\widehat{\mathbf{x}} \quad , \tag{E.10}$$

where $\mathbf{g}$ is the vector of first-order derivatives and $\mathbf{H}$ is the Hessian of $F(\mathbf{x})$. This scheme gives a good estimate of the function value at the maximum location when the true maximum is very close to the central pixel in a 3×3 neighbourhood. If instead, the true maximum is midway between the central pixel and one of the surrounding 8 pixels, there is not enough support within $\mathbf{F}$ to approximate $F(\mathbf{x})$ accurately.

The estimated value of $F(\widehat{\mathbf{x}})$ will have least error when the true maximum (or our current estimate of the true maximum) is at the centre of the group of pixels used to form $\mathbf{F}$. We use a 4×4 neighbourhood such that the first estimate of the maximum is one of the four central pixels. We get the first sub-pixel estimate, $\widehat{\mathbf{x}}_1$ of the maximum location using this 4×4 neighbourhood. We use a bi-linear weighting scheme for the weights from the four 9-point weightings

Distances of $\widehat{\mathbf{x}}_1$ from each of the four central pixels are used to assign weights to the four 3×3 subset of the 4×4 neighbourhood. Each 3×3 subset is centred on one of the four central pixels. The weight matrix $\mathbf{W}$ is a weighted mixture of the four 3×3 weightings

$$\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \tag{E.11}$$

Therefore, the distances of $\widehat{\mathbf{x}}_1$ from each of the four central pixels determines how much of each of the corresponding 3×3 neighbourhood will be used to form the revised $\mathbf{F}$. A quadratic weighting of the pixel values ensures that the pixels closest to $\widehat{\mathbf{x}}$ have a greater effect on the approximation than the pixels away from $\widehat{\mathbf{x}}$.

To illustrate the weighting scheme, we show three examples in Figure E.1

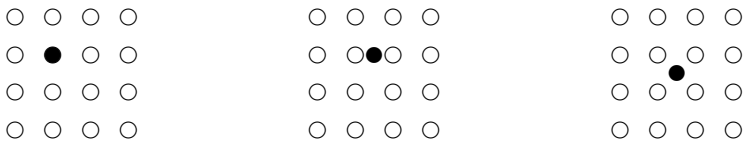


Figure E.1: True maximum located at (a): one of the four central pixels, leads to weights $\mathbf{W}_a$ (b): the midpoint of two of the four central pixels leads to weights $\mathbf{W}_b$. (c): the midpoint of the four central pixels leads to weights $\mathbf{W}_c$.

and list the weights corresponding to cases (a)-(c) here

$$\mathbf{W}_a = \frac{1}{16}\begin{bmatrix} 1 & 2 & 1 & 0 \\ 2 & 4 & 2 & 0 \\ 1 & 2 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \mathbf{W}_b = \frac{1}{32}\begin{bmatrix} 1 & 3 & 3 & 1 \\ 2 & 6 & 6 & 2 \\ 1 & 3 & 3 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \mathbf{W}_c = \frac{1}{64}\begin{bmatrix} 1 & 3 & 3 & 1 \\ 3 & 9 & 9 & 3 \\ 3 & 9 & 9 & 3 \\ 1 & 3 & 3 & 1 \end{bmatrix} \quad \text{(E.12)}$$

The expressions for the location of the maximum and the corresponding function value at iteration τ are obtained by replacing $\mathbf{F}$ with $\mathbf{F}_{\tau-1} = \mathbf{W}_{\tau-1} \cdot \mathbf{F}$ ($\cdot$ denotes an element-wise multiplication), in (E.8)

$$F(\mathbf{x}) = \mathbf{W}\left[F + \mathbf{g}^\top \cdot \mathbf{x} + \frac{1}{2}\mathbf{x}^\top \mathbf{H}\mathbf{x}\right] \quad , \quad \text{(E.13)}$$

$$\widehat{\mathbf{x}}_\tau = -\mathbf{H}_{\tau-1}^{-1}\mathbf{g}_{\tau-1} \quad \text{and} \quad \text{(E.14)}$$

$$F(\widehat{\mathbf{x}}_\tau) = \mathbf{F}_{\tau-1} + \frac{1}{2}\mathbf{g}_{\tau-1}^\top \widehat{\mathbf{x}}_\tau \quad \text{(E.15)}$$

where $\mathbf{g}_\tau$ and $\mathbf{H}_\tau$ is calculated over $\mathbf{F}_{\tau-1}$ subject to initial condition $\mathbf{F}_0 = \mathbf{F}$. The process is repeated until $\|\widehat{\mathbf{x}}_\tau - \widehat{\mathbf{x}}_{\tau-1}\|$ is within an acceptable error limit. The results of using this maximum interpolation process are seen in Figure E.2.

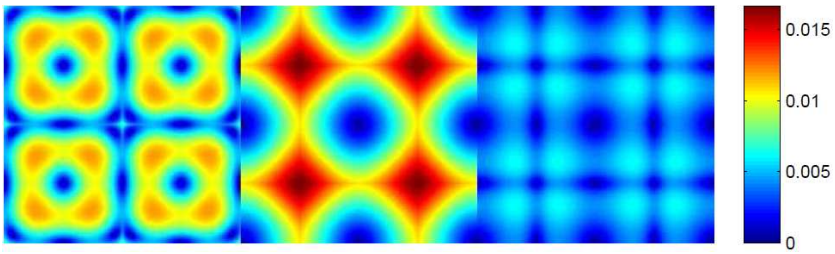


Figure E.2: Comparison of the localisation error of three maxima interpolation methods: The localisation error of (Left) 3×3 fixed weight maxima interpolation method, (Middle) 4×4 fixed weight maxima-interpolation method and (Right) 4×4 adaptive maxima interpolation method is plotted. The colour indicates the error, red = high error, blue = low error. The true maximum is shifted within the range [-1,1] pixels in both x and y direction. The input image is an image of a blob with radius = 5 pixels and width of the ramp = 3 pixels. The 4×4 adaptive weight maxima interpolation scheme clearly has much lower error as compared to the other two methods.

Acknowledgements: Software

Software from the following sources is gratefully acknowledged:

Author - Software reference
N. Kingsbury - DTCWT methods [Kingsbury, 2001, 2006]
N. Kingsbury, J. Fauqueur - FKA keypoint detector [Fauqueur et al., 2006]
D. Lowe - SIFT detector and descriptor [Lowe, 2004]
A. Vedaldi - SIFT detector and descriptor (MATLAB [Vedaldi, 2007])
K. Mikolajczyk, T. Tuytelaars - Har-Aff/Hes-Aff/IBR [Mikolajczyk, 2005]
B. Triggs - Assorted camera pose routines [Triggs, 1999]
B. Triggs - Multiscale epipolar geometry [Triggs and Bendale, 2010]

Acronyms

DTCWT	Dual-Tree Complex Wavelet Transform
4S-DTCWT	Four Scale Dual-Tree Complex Wavelet Transform
BTK	Bendale - Triggs - Kingsbury (detector/descriptor)
FKA	Fauqueur - Kingsbury - Anderson (detector)
SIFT	Scale Invariant Feature Transform
DoG	Difference of Gaussian
MSER	Maximally Stable Extremal Regions
IBR	Intensity Based Regions
EBR	Edge Based Regions
HAR–AFF	Harris Affine
HES–AFF	Hessian Affine
RANSAC	Random Sample Consensus
FFT	Fast Fourier Transform
IFFT	Inverse Fast Fourier Transform
PCA	Principal Components Analysis
DOLP	Difference of Low Pass Transform

Bibliography

J. Babaud, A. P. Witkin, M. Baudin, and R. O. Duda. Uniqueness of the gaussian kernel for scale-space filtering. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 8(1):26–33, 1986. 10

A. Baumberg. Reliable feature matching across widely separated views. In *IEEE Conference Computer Vision & Pattern Recognition (CVPR)*, pages 1774–1781, 2000. 14

H. Bay, T. Tuytelaars, and L. J. Van Gool. SURF: Speeded up robust features. In *European Conference on Computer Vision (ECCV)*, pages 404–417, 2006. 13

H. Bay, A. Ess, T. Tuytelaars, and L. J. Van Gool. SURF: Speeded up robust features. *Computer Vision and Image Understanding (CVIU)*, 110 (3):346–359, 2008. 13, 18

P. R. Beaudet. Rotationally invariant image operators. In *International Joint Conference on Pattern Recognition*, pages 579–583, 1978. 14

J. Beis and D. Lowe. Shape indexing using approximate nearest-neighbour search in high dimensional spaces. In *IEEE Conference Computer Vision & Pattern Recognition (CVPR)*, pages 1000–1006, 1997. 13

S. Belongie, J. Malik, and J. Puzicha. Shape context: A new descriptor for shape matching and object recognition. In *Neural Information Processing Systems (NIPS)*, 2000. 17

S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 24(4):509–522, 2002. 17

P. Bendale, J. D. B. Nelson, and N. Kingsbury. Techniques for establishing keypoint correspondences via polar matching with complex wavelets. Unpublished manuscript, originally submitted to IEEE Conference Computer Vision & Pattern Recognition (CVPR) 2008, November 2007. 76

P. Bendale, J. Cameron, B. Triggs, and N. Kingsbury. A calibrated 3D dataset for automatic evaluation of keypoint detectors and descriptors. Technical Report CUED/F-INFENG/TR. 655, Cambridge University Engineering Department, Cambridge, UK, August 2010a. 83, 139, 142

P. Bendale, B. Triggs, and N. Kingsbury. Multiscale keypoint analysis based on complex wavelets. In *British Machine Vision Conference (BMVC)*, 2010b. 29, 99, 100, 108, 127

A. Berg and J. Malik. Geometric blur for template matching. In *IEEE Conference Computer Vision & Pattern Recognition (CVPR)*, pages 607–614, 2001. 17

M. Brown. *Multi-Image Matching Using Invariant Features*. PhD thesis, University of British Columbia, 2005. 25, 43

M. Brown and D. Lowe. Invariant features from interest point groups. In *British Machine Vision Conference (BMVC)*, pages 656–665, 2002. 11

M. Brown and D. Lowe. Automatic panoramic image stitching using invariant features. *International Journal Computer Vision (IJCV)*, 74(1):59–73, 2007. 4

M. Brown, G. Hua, and S. Winder. Discriminative learning of local image descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 33(1):43–57, 2011. 17, 18

C. Cabani and W. J. MacLean. Implmementation of an affine-covariant feature detector in field-programmable gate arrays. In *International Conference on Computer Vision Systems*, 2007. 19

CalTech. Caltech motorbikes (side) database, 2001. 2

G. Carneiro and A. Jepson. Flexible spatial configuration of local image features. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 29(12):2089–2104, 2007. 5, 24, 28

O. Chum and J. Matas. Matching with PROSAC - Progressive Sample Consensus. In *IEEE Conference Computer Vision & Pattern Recognition (CVPR)*, pages 220–226, 2005. 77

D. Coffin. DCRAW: Decoding raw digital photos in linux. URL: http://cybercom.net/~dcoffin/dcraw, 2008. 81

D. Comaniciu. An algorithm for data-driven bandwidth selection. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 25 (2):281–288, 2003. 155

D. Comaniciu, V. Ramesh, and P. Meer. The variable bandwidth mean shift and data-driven scale selection. In *IEEE International Conference on Computer Vision (ICCV)*, pages 438–445, 2001. 155

J. Crowley and A. Parker. A representation for shape based on peaks and ridges in the difference of low pass transform. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 6(2):156–170, 1984. 43

N. Dalal. *Finding People in Images and Videos*. PhD thesis, Institut National Polytechnique de Grenoble, INRIA Rhône-Alpes, Grenoble, July 2006. 155

J. Davis and M. Goadrich. The relationship between precision-recall and roc curves. In *International Conference Machine Learning (ICML)*, pages 233–240, 2006. 26

O. Faugeras, Q.-T. Luong, and S.J. Maybank. Camera self calibration: Theory and experiments. In *European Conference on Computer Vision (ECCV)*, 1992. 101, 102

J. Fauqueur, N. Kingsbury, and R. Anderson. Multiscale keypoint detection using the dual-tree complex wavelet transform. In *International Conference Image Processing (ICIP)*, 2006. 5, 23, 28, 41, 43, 44, 83, 118, 161

M. Fischler and R. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the Association for Computing Machinery*, 24(6):381–395, 1981. 77

A. Fitzgibbon. Personal communication, 2010. 143

A. Fitzgibbon, G. Cross, and A. Zisserman. Automatic 3D model construction for turn-table sequences. In R. Koch and L. J. Van Gool, editors, *3D Structure from Multiple Images of Large-Scale Environments*, pages 155–170. Springer-Verlag, 1998. 82

P-E. Forssén and A. Moe. Blobs in epipolar geometry. In *SSBA Symposium on Image Analysis*, pages 82–85, 2004. 100, 101

W. Förstner. A framework for low-level feature extraction. In *European Conference on Computer Vision (ECCV)*, pages 383–394, 1994. 40

F. Fraundorfer and H. Bischof. A novel performance evaluation method of local detectors on non-planar scenes. In *IEEE Conference Computer Vision & Pattern Recognition (CVPR) - Workshops*, volume 03, 2005. 26, 80, 83, 100

W. Freeman and E. Adelson. The design and use of steerable filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 13 (9):891–906, 1991. 20

M. Frigo and S. Johnson. The design and implementation of FFTW3. *Proceedings of the IEEE*, 93(2):216–231, 2005. Special issue on "Program Generation, Optimization, and Platform Adaptation". 74

K. Grauman and T. Darrell. The pyramid match kernel: Discriminative classification with sets of image features. In *IEEE International Conference on Computer Vision (ICCV)*, 2005. 77

C. Harris and M. J. Stephens. A combined corner and edge detector. In *Alvey Vision Conference*, pages 147–152, 1988. 12, 14, 40

R. Hartley. In defense of the eight-point algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 19(6):580–593, 1997a. 124

R. Hartley. Lines and points in three views and the trifocal tensor. *International Journal Computer Vision (IJCV)*, 22(2):125–140, 1997b. 125

R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision.* Cambridge University Press, second edition, 2003. 100, 101, 102, 121, 124, 125

S. Heymann, K. Maller, A. Smolic, B. Froehlich, and T. Wiegand. SIFT implementation and optimization for general-purpose GPU. In *International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision*, 2007. 19

G. Hua, M. Brown, and S. Winder. Discriminant embedding for local image descriptors. In *IEEE International Conference on Computer Vision (ICCV)*, 2007. 17

D. Hubel. *Eye, Brain and Vision.* W. H. Freeman, second edition, 1995. 19

D. Hubel, T. Wiesel, and M. Stryker. Orientation columns in macaque monkey visual cortex demonstrated using 2-deoxyglucose autoradiographic technique. *Nature*, 269(5626):328–330, 1977. 19

S. G. Johnson and M. Frigo. Implementing FFTs in practice. In C. Sidney Burrus, editor, *Fast Fourier Transforms*, chapter 11. Connexions, Rice University, Houston TX, September 2008. 74

F. Kahl and A. Heyden. Using conic correspondence in two images to estimate the epipolar geometry. In *IEEE International Conference on Computer Vision (ICCV)*, 1998. 101

Y. Ke and R. Sukthankar. PCA-SIFT: A more distinctive representation for local image descriptors. In *IEEE Conference Computer Vision & Pattern Recognition (CVPR)*, 2004. 13

N. Kingsbury. Image processing with complex wavelets. *Philosophical Transactions of the Royal Society London A*, September 1999. Discussion Meeting on "Wavelets: the key to intermittent information?", London, February 24-25, 1999. 21

N. Kingsbury. Complex wavelets for shift invariant analysis and filtering of signals. *Journal of Applied and Computational Harmonic Analysis*, 10(3): 234–253, 2001. 19, 20, 21, 161

N. Kingsbury. Rotation-invariant local feature matching with complex wavelets. In *European Conference Signal Processing (EUSIPCO)*, 2006. 5, 21, 22, 23, 24, 28, 65, 66, 69, 71, 72, 161

J. Koenderink. The structure of images. *Biological Cybernetics*, 50:363–396, 1984. 10

J. Koenderink and A. J. Van Doorn. Representation of local geometry in the visual system. *Biological Cybernetics*, 55:367–375, 1987. 24

M. Leordeanu and M. Hebert. A spectral technique for correspondence problems using pairwise constraints. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1482–1489, 2007. 77

V. Lepetit and P. Fua. Keypoint recognition using randomized trees. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 28 (9):1465–1479, 2006. 4, 18

T. Lindeberg. Detecting salient blob-like image structures and their scales with a scale-space primal sketch: a method for focus-of-attention. *International Journal Computer Vision (IJCV)*, 11(3):283–318, 1993. 43

T. Lindeberg. *Scale-space theory in computer vision.* Kluwer Academic Publishers, 1994. ISBN 0-7923-9418-6. Published by Springer later as ISBN 978-0-7923-9418-1. 10

T. Lindeberg. Direct estimation of affine image deformation using visual front-end operations with automatic scale selection. In *IEEE International Conference on Computer Vision (ICCV)*, pages 134–141, 1995. 14

T. Lindeberg. Feature detection with automatic scale selection. *International Journal Computer Vision (IJCV)*, 30(2):77–116, 1998. 14, 24, 43, 55

D. Lowe. Object recognition from local scale-invariant features. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1150–1157, 1999. 2, 5, 9, 28, 37

D. Lowe. Local feature view clustering for 3D object recognition. In *IEEE Conference Computer Vision & Pattern Recognition (CVPR)*, pages 682–688, 2001. 4

D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal Computer Vision (IJCV)*, 60(2):91–110, 2004. 2, 9, 25, 35, 43, 83, 93, 100, 161

Q. Luong and O. Faugeras. The fundamental matrix: Theory, algorithms, and stability analysis. *International Journal Computer Vision (IJCV)*, 17 (1):43–75, 1996. 124

D. Marr. *Vision: A computational investigation into the human representation and processing of visual information.* W. H. Freeman and Company, San Francisco, CA, USA, 1982. 9, 10

D. Marr and E. Hildreth. Theory of edge detection. *Proceedings of the Royal Society of London Series B*, 207:187–217, 1980. 10

J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *British Machine Vision Conference (BMVC)*, pages 384–393, 2002. 15, 16

K. Mikolajczyk. *Detection of Local Features Invariant to Affine Transformations, Application to Matching and Recognition.* PhD thesis, Institut National de Polytechniques de Grenoble, France, 2002. 25, 55

K. Mikolajczyk. Affine covariant features. URL: http://www.robots.ox.ac.uk/~vgg/research/affine/index.html, 2005. 26, 93, 161

K. Mikolajczyk and C. Schmid. Indexing based on scale invariant interest points. In *IEEE International Conference on Computer Vision (ICCV)*, pages 525–531, 2001. 4, 24, 55

K. Mikolajczyk and C. Schmid. Scale and affine invariant interest point detectors. *International Journal Computer Vision (IJCV)*, 60(1):63–86, 2004. 14, 93, 100

K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 27(10):1615–1630, 2005. 2, 17, 24, 25, 80

K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. J. Van Gool. A comparison of affine region detectors. *International Journal Computer Vision (IJCV)*, 65(1–2):43–72, 2005. 5, 25, 26, 27, 28, 43, 55, 68, 83, 100

P. Moreels and P. Perona. Evaluation of features detectors and descriptors based on 3D objects. In *IEEE International Conference on Computer Vision (ICCV)*, volume 1, pages 800–807, 2005. 26, 27, 81

P. Moreels and P. Perona. Evaluation of features detectors and descriptors based on 3D objects. *International Journal Computer Vision (IJCV)*, 73 (3):263–287, 2007. 26, 27, 80, 81, 83, 100

E. Mortensen, H. Deng, and L. Shapiro. A SIFT descriptor with global context. In *IEEE Conference Computer Vision & Pattern Recognition (CVPR)*, 2005. 13, 77

L. Mueller et al. gPhoto2: A free, redistributable, ready to use set of digital camera software applications for unix-like systems. URL: http://www.gphoto.org/, 2000. 139

J. Nelson and N. Kingsbury. Enhanced shift and scale tolerance for rotation invariant polar matching with dual-tree wavelets. *IEEE Transactions on Image Processing*, PP(99):1, 2010. doi: 10.1109/TIP.2010.2069711. 118

E. S. Ng and N. Kingsbury. Matching of interest point groups with pairwise spatial constraints. In *International Conference Image Processing (ICIP)*, 2010. 77

Directed Perception. PTU D46: A miniature pan-tilt unit for accurate real-time positioning of cameras. User manual and command reference (Version 2.14.0) URL: http://www.dperception.com/pdf/products/PTU-D46/PTU-manual-d46.pdf, 2006. 139

J. Porrill and S. Pollard. Curve matching and stereo calibration. *Image and Vision Computing*, 9(1):45–50, 1991. 101, 102, 103

E. Rosten and T. Drummond. Fusing points and lines for high performance tracking. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1508–1511, 2005. 18

E. Rosten and T. Drummond. Machine learning for high-speed corner detection. In *European Conference on Computer Vision (ECCV)*, pages 430–443, 2006. 18

E. Rosten, R. Porter, and T. Drummond. FASTER and better: A machine learning approach to corner detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 32(1):105–119, 2010. 4, 18

F. Schaffalitzky and A. Zisserman. Multi-view matching for unordered image sets. In *European Conference on Computer Vision (ECCV)*, 2002. 4, 24

C. Schmid and R. Mohr. Local grayvalue invariants for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 19(5):530–535, 1997. 2, 4, 24, 25, 77, 80

I. W. Selesnick, R. G. Baraniuk, and N. G. Kingsbury. The dual-tree complex wavelet transform. *IEEE Signal Processing Magazine*, 22(6):123–151, 2005. 21

A. Shashua. Trilinearity in visual recognition by alignment. In *European Conference on Computer Vision (ECCV)*, pages 479–484, 1994. 125

A. Shashua. Algebraic functions for recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 17(8):779–789, August 1995. 125

E. Simoncelli and W. Freeman. The steerable pyramid: a flexible architecture for multi-scale derivative computation. In *International Conference Image Processing (ICIP)*, 1995. 19, 20

S. N. Sinha, J. M. Frahm, M. Pollefeys, and Y. Genc. GPU-based video feature tracking and matching. In *EDGE - Workshop on Edge Computing Using New Commodity Architectures*, 2006. 19

S. M. Smith and J. M. Brady. SUSAN - A new approach to low level image processing. *International Journal Computer Vision (IJCV)*, 23(34):45–78, 1997. 18

M. Spetsakis and J. Aloimonos. A multi-frame approach to visual motion perception. *International Journal Computer Vision (IJCV)*, 16(3):245–255, 1991. 125

K. Strobl, W. Sepp, S. Fuchs, C. Paredes, and K. Arbter. DLR CalDe and DLR CalLab. Institute of Robotics and Mechatronics, German Aerospace Center (DLR), Oberpfaffenhofen, Germany. URL: http://www.robotic.dlr.de/callab/, 2005. 4, 81, 139, 141

Richard Szeliski. *Computer Vision: Algorithms and Applications*. Springer (In preparation), draft version dated spetember 3, 2010 edition, 2010. An electronic draft for non-commercial personal use only is available at http://szeliski.org/Book/. 26

E. Tola, V. Lepetit, and P. Fua. A fast local descriptor for dense matching. In *IEEE Conference Computer Vision & Pattern Recognition (CVPR)*, pages 1–8, 2008. 18

B. Triggs. Matching constraints and the joint image. In *IEEE International Conference on Computer Vision (ICCV)*, pages 338–343, 1995. 125

B. Triggs. Assorted camera pose routines. URL: http://ljk.imag.fr/membres/Bill.Triggs/src/pose.tar.gz, 1999. 161

B. Triggs. Joint feature distributions for image correspondence. In *IEEE International Conference on Computer Vision (ICCV)*, pages 201 –208, 2001. 77

B. Triggs. Detecting keypoints with stable position, orientation and scale under illumination changes. In *European Conference on Computer Vision (ECCV)*, 2004. 100

B. Triggs and P. Bendale. Epipolar constraints for multiscale matching. In *British Machine Vision Conference (BMVC)*, 2010. 99, 103, 161

T. Tuytelaars and L. J. Van Gool. Matching widely separated views based on affine invariant regions. *International Journal Computer Vision (IJCV)*, 59(1):61–85, 2004. ISSN 0920–5691. 16, 17, 83

A. Vedaldi. An open implementation of the SIFT detector and descriptor. Technical Report 070012, UCLA CSD, 2007. 161

A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms. URL: http://www.vlfeat.org/, 2008. 13, 96

L. Vincent and P. Soille. Watersheds in digital spaces: an efficient algorithm based on immersion simulations. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 13(6):583–598, 1991. 15

P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *IEEE Conference Computer Vision & Pattern Recognition (CVPR)*, pages 511–518, 2001. 13

J. Weng, N. Ahuja, and T. Huang. Closed-form solution and maximum likelihood: A robust approach to motion and structure estimation. In *IEEE Conference Computer Vision & Pattern Recognition (CVPR)*, 1988. 125

S. Winder and M. Brown. Learning local image descriptors. In *IEEE Conference Computer Vision & Pattern Recognition (CVPR)*, 2007. 17, 46, 67

S. Winder, G. Hua, and M. Brown. Picking the best DAISY. In *IEEE Conference Computer Vision & Pattern Recognition (CVPR)*, pages 178–185, 2009. 17, 18

A. Witkin. Scale-space filtering. In *International Joint Conference Artificial Intelligence*, pages 1019–1022, 1983. 10

G. Xu and Z. Zhang. *Epipolar geometry in stereo, motion and object recognition.* Kluwer Academic Publishers, 1996. 124